



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Pasi Ojala

LAAJAKAISTATILAUKSEN PERUMI- SENAUTOMATISOINTI

Liiketalouden yksikkö
2016

TIIVISTELMÄ

Tekijä	Pasi Ojala
Opinnäytetyön nimi	Laajakaistatilauksen Perumisen automatisointi
Vuosi	2016
Kieli	Suomi
Sivumäärä	58
Ohjaaja	Sirkka Hellman

Opinnäytetyöni aiheena oli kehittää laajakaistatoimituksen hallintajärjestelmässä olevaa laajakaistatilauksen perumisen automatisointitoimintoa. Toiminnallinen osuus koostui kaavioiden sekä sanallisten käyttötapauksen luomisesta.

Teoriaosuudessa käyn läpi ohjelmistotuotantoa sekä UML-kaavioista tilakaavion, aktiviteettikaavion, sekvenssikaavion ja käyttötapauskaavion sekä näihin liittyviä tärkeitä osa-alueita. Näiden lisäksi kerron laadusta ja tiedon oikeellisuuden merkityksestä järjestelmässä ja roboautomaatiikasta.

Empiirisessä osassa kerron laajakaistatilauksen toimitus- ja automaatioprosessista sekä perumisen nykyhetkestä. Lisäksi käydään läpi, miten perutut tapaukset tunnistettiin, mitkä tapaukset valittiin automatisoinnin piiriin sekä lopuksi kerron kaavioiden luonnista ja sanallisten käyttötapauksen teosta. Toimeksiantajalle toteutettuja kaavioita ja sanallisia käyttötapauksia ei tässä opinnäytetyössä esitetä.

Työn tuloksena kaaviot sekä sanalliset käyttötapaukset ovat valmiina työmääräarviota sekä mahdollista ohjelmointia ja testausta varten. Automatisoinnin mahdollinen käyttöönotto tapahtuu vasta tämän opinnäytetyön jälkeen, joten automatisoinnin hyötyjä ei käydä tässä opinnäytetyössä läpi.

ABSTRACT

Author	Pasi Ojala
Title	Automated Cancellation of a Broadband Order
Year	2016
Language	Finnish
Pages	58
Name of Supervisor	Sirkka Hellman

The thesis subject was to develop order cancelling automation for a broadband delivery software program. The goal was to create diagrams and verbal use cases.

In the theoretical study of the thesis software engineering and UML diagrams called state machine diagrams, activity diagrams, sequence diagrams, use case diagrams as well as important elements of these are described. Also the importance of quality and accuracy of information as well as robot automation are explained.

As a part of the empirical study it is described how broadband delivery and process of automation work, how order cancelling automation works at the moment, how cancelled orders are recognized and which cases are selected as the scope of automation. Also creating diagrams and verbal use cases are explained. The diagrams and verbal use cases which were made for client are not presented in this thesis.

As a result of the work the diagrams and verbal use cases are ready for workload estimation and possible programming and testing. Possible commissioning of automation will take place after this thesis is completed so the benefits of automation are not a part of this work.

Keywords	Software engineering, automation, cancelling an order, UML-diagrams, verbal use cases
----------	---

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

OPINNÄYTETYÖSSÄ ESIINTYVÄT KÄSITTEET	4
1 JOHDANTO.....	6
1.1 Opinnäytetyön tausta	6
2 LAAJAKAISTAINEN INTERNETYHTEYS	8
2.1 Laajakaistaliittymä.....	8
2.2 Laajakaistayhteys	8
2.2.1 Kiinteä laajakaistayhteys.....	9
2.2.2 Laajakaistayhteys tekniikoita.....	9
2.3 Mobiililaajakaistayhteys	10
2.4 Satelliittilaajakaistayhteys.....	11
3 OHJELMISTOTUOTANNON KÄYTÄNTÖJÄ.....	12
3.1 Ohjelmistotuotanto.....	12
3.2 Ohjelmiston uudistaminen	12
3.3 Ohjelmiston testaus	13
3.4 Ohjelmiston ylläpito.....	15
4 UML	17
4.1 UML ohjelmistotuotannossa.....	17
4.2 UML-kaaviot.....	18
4.3 Käyttäjätarinat.....	19
4.4 Käyttötapaukset.....	20
4.5 Käyttötapauskaavio	21
4.6 Sanalliset käyttötapaukset	23
4.7 Tilakaavio	24
4.8 Aktiviteettikaavio.....	27
4.9 Sekvenssikaavio.....	31
5 LAADUN JA TIEDON OIKEELLISUUDEN MERKITYS	
JÄRJESTELMÄSSÄ	35
5.1 Laatu	35

5.2	Laadun määrittely	36
5.3	Laatujärjestelmä	36
5.4	Laadunvarmistus	37
5.5	Laadun mittaaminen.....	38
6	OHJELMISTOROBOTIIKKA TIETOTYÖSSÄ	41
7	OPINNÄYTETYÖN TOIMINNALLINEN OSUUS	44
7.1	Opinnäytetyön tavoite	44
7.2	Tilauksen toimitus -ja automaatioprosessi.....	44
7.3	Laajakaistatilauksen peruminen nykyhetki.....	46
7.4	Tapausten tunnistaminen perumisen automatisoinnin laajentamiseksi ..	49
7.5	Perumisen automatisoinnin piiriin valitut tapaukset.....	51
7.6	Kaavioiden ja sanallisten käyttötapauksen luonti	52
8	JOHTOPÄÄTÖKSET JA POHDINTA	54
	LÄHTEET	55

KUVIO- JA TAULUKKOLUETTELO

Kuvio 1 UML-kaaviot (Seidl ym. 2015, 15.).....	18
Kuvio 2 Käyttötapauskaavio (Haikala & Mikkonen 2011, 77)	22
Kuvio 3 Sanallinen käyttötapaus (Haikala & Mikkonen 2011, 80)	24
Kuvio 4 Tilakaavion symboleja (Seidl ym. 2015, 106)	26
Kuvio 5 Tilakaavio (Wikispaces, 2016).....	27
Kuvio 6 Aktiviteettikaavion symboleja (Seidl ym. 2015, 165)	29
Kuvio 7 Aktiviteettikaavio 1	30
Kuvio 8 Aktiviteettikaavio 2.....	31
Kuvio 9 Sekvenssikaavion symboleja (Seidl ym. 2015, 140).....	33
Kuvio 10 Sekvenssikaavio (Haikala & Mikkonen 2011, 98)	34
Kuvio 11 Näkymä kiinteän verkon dokumentointijärjestelmästä.....	45
Kuvio 12 Laajakaistatilauksen peruminen tilaus rivien ollessa UU/TO-tilassa....	48
Kuvio 13 Tunnistetut tapaukset	49
Kuvio 14 Toimituksen hallinnointijärjestelmästä näkymä 1	50
Kuvio 15 Toimituksen hallinnointijärjestelmästä näkymä 2	50
Kuvio 16 Toimituksen hallinnointijärjestelmästä näkymä 3	51
Kuvio 17 Automatisoinnin piiriin valitut tapaukset.....	51

OPINNÄYTETYÖSSÄ ESIINTYVÄT KÄSITTEET

WLAN	Wireless Local Area Network, langaton lähiverkko.
DSL	Digital Subscriber Line, jonka yhteystyyppin nopeus voi olla epäsymmetrinen tai symmetrinen.
ADSL	Asymmetric Digital Subscriber Line, jonka yhteystyyppi on epäsymmetrinen.
ADSL2+	Asymmetric Digital Subscriber Line 2, epäsymmetrinen yhteystyyppi, joka mahdollistaa maksimissaan 24M tiedonsiirto nopeuden.
VDSL2	Very High Speed Subscriber Line 2, epäsymmetrinen DSL-yhteys
GSM	Global System for Mobile Communications on toisen sukupolven matkaviestinjärjestelmä.
GPRS	General Packet Radio Service, GSM-verkossa toimiva datapalvelu, jota pääasiassa käytetään langattoman verkkoyhteyden muodostuksessa.
EDGE	Enhanced Data Rates for Global Evolution, joka perustuu GPRS-yhteyteen, mutta datasiirtotekniikka on kolme kertaa nopeampi.
UML	Unified Modeling Language on graafinen mallinnuskieli, joka sisältää 14 erilaista mallinnus kaaviota.
RPA	Robotic Process Automation on ohjelmistorobotiikan sovellustaso.
SPA	Smart Process Automation on ohjelmistorobotiikan sovellustaso.
JST	Job Scheduling tool on alihankkijan asennusajanvaraus järjestelmä.
UU-tila	Työtä/riviä ei ole otettu käsittelyyn.
TO-tila	Työ/rivi on siirretty työn alla tilaan.

LAH-tila	Rivi odottaa vastausta toisesta järjestelmästä.
VA-tila	Rivi on käsitelty ja valmistunut.
SII-tila	Rivi on käsitelty ja siirtynyt valmiiksi.
EPA-tila	Rivin käsittely on epäonnistunut.

1 JOHDANTO

Opinnäytetyöni aiheena on kehittää olemassa olevaan kiinteänlaajakaistatoimitusten hallintajärjestelmään perutun laajakaistatilauksen käsittelyn automatisointi. Työn kuvana on mallintaa kaavioilla sekä sanallisilla käyttötapauksilla automaatio toiminto, jossa peruttu laajakaistatilaus käsitellään automaation toimesta. Kaavioiden sekä sanallisten käyttötapauksien kuvausten perusteella toimittaja tekee työmääräarvioinnin ja mahdollisen toteutuksen. Opinnäytetyöstä on rajattu koodaus ja testaus.

1.1 Opinnäytetyön tausta

Toimeksiantajana toimii TeliaSonera Finland Oy. TeliaSonera Finland Oy on Suomessa toimiva verkko-operaattori, joka tarjoaa yrityksille, yhteisöille ja kuluttajille viestintä- televisio- ja tietoliikennepalveluja. (TeliaSonera, 2013.)

Perutun laajakaistatilauksen automaattisen käsittelyn tarve havaittiin, kun TeliaSonera Finland Oy käynnisti vuonna 2012 projektin, jonka tarkoituksena on parantaa kiinteän verkkotietojärjestelmän datan laatua ja oikeellisuutta.

Projektissa korjattiin kuluttajalaajakaistojen toimituksen yhteydessä tapahtuneita tietosisältövirheitä, jotka kohdistuvat kiinteän verkon dokumentointi- sekä verkon hallintajärjestelmään. Korjauksissa havaittiin, että yksi suurimmista syistä verkkotiedon virheellisyyteen tulee inhimillisistä virheistä ja varsinkin virheellisesti peruttujen laajakaistatilauksien yhteydessä.

Virheellisesti perutut laajakaistatilaukset aiheuttavat tietosisältövirheitä järjestelmien välille ja järjestelmiin jää ylimääräisiä laajakaistayhteyksiä. Nämä ylimääräiset yhteydet aiheuttavat uusien laajakaistaliittymien automaattireititysten epäonnistumisia, varaavat verkkolaitteilta kapasiteettia verkosta, aiheuttavat kuluja turhaan tilatuista vuokrajohdoista sekä verkon muutostöissä aiheutuu epäselvyyksiä muutostyön alle kuuluvista liittymistä ja niiden määristä.

Inhimillisten virheiden ja tilausten peruutusten työllistävästä vaikutuksesta johtuen haluttiin kehittää perumisen automatiikka siten, että käyttäjän ei tarvitse koskea peruttuun tilaukseen.

2 LAAJAKAISTAINEN INTERNETYHTEYS

Opinnäytetyössäni kiinteiden laajakaistaliittymätilausten automatisointi on keskeisessä osassa. Automatisointi käsittää kiinteistä laajakaistayhteystekniikoista ADSL-, VDSL-, Ethernet- ja Valokuidun. Tämän kappaleen tarkoituksena on antaa lukijalle ymmärrys siitä, mikä laajakaistayhteys on, millaisia vaihtoehtoja internetiin pääsyyn löytyy nykypäivänä sekä millaisia laajakaistayhteys tekniikoita ihmiset nykypäivänä voivat käyttää.

2.1 Laajakaistaliittymä

Laajakaistaliittymä on asiakkaalle toimitettava sopimuksellinen ja teknillinen kokonaisuus. Toimitettavan liittymän avulla tilaaja saa muodostettua yhteyden internetiin tilatulla tiedonsiirtonopeudella. Asiakkaat voivat tilata liittymän ohien lisäpalveluita, kuten laajakaistatelevision(IPTV). Laajakaistaliittymä pystytään toteuttamaan sekä langattomasti että kiinteästi. (Sanastokeskus TSK, 2012.)

2.2 Laajakaistayhteys

Laajakaistayhteys on tietoliikenneyhteys, jossa käyttäjälle siirtyy tietoliikenneverkosta tilatulla tiedonsiirtonopeudella dataa. Laajakaistayhteyksillä on enimmäisnopeus ja vähimmäisnopeus riippuen tilatun yhteyden nopeudesta. Enimmäisnopeudella tarkoitetaan suurinta tiedonsiirtonopeutta, jossa käyttäjältä tai käyttäjälle siirretään dataa tietoliikenneverkosta verkkoon. Vähimmäisnopeus puolestaan on tiedonsiirtonopeus, jonka palveluntarjoaja lupaa tilaajalle vähintään saatavaksi. (Sanastokeskus TSK, 2012.) Nettiyhteys katsotaan laajakaistayhteydeksi, jos sen nopeusluokka on vähintään 256kb/s (Tietoliikenteen ja tietotekniikan keskusliitto, FiCom ry 2016).

Laajakaistayhteyksiä pystytään toteuttamaan kiinteästi tai langattomasti. Laajakaistayhteydellä saadaan muodostettua yhteys internetiin, joka on toisiinsa yhdistettyjen verkkojen kokonaisuus. Internet koostuu tietokoneista, jotka ovat liitettyinä toisiinsa sekä tietoverkoista, joiden avulla yhdistetään maailman tietoverkot.

Internet muodostuu monista tietokone- ja tiedonsiirtojärjestelmistä. (Tietoliikenteen ja tietotekniikan keskusliitto, FiCom ry 2016.)

2.2.1 Kiinteä laajakaistayhteys

Kiinteä laajakaistayhteys on tietoliikennekaapeleiden avulla yhdistettävä laajakaistayhteys. Kiinteän laajakaistayhteyden toteutuksessa voidaan käyttää esim. yleiskaapelointia. Kiinteitä laajakaistayhteyksiä toteutetaan eri päätelaitteilla ja tekniikoilla, kiinteästi ja langattomasti. Kiinteitä laajakaistayhteystekniikoita on erilaisia, joita ovat DSL-yhteys, ADSL-yhteys, VDSL-yhteys, valokuituyhteys, kaapelimodeemiyhteys ja ethernet-yhteys. (Sanastokeskus TSK, 2012.) Kiinteää laajakaistayhteyttä voidaan käyttää ainoastaan paikassa, johon se on tilattu ja se voidaan muodostaa tilaajan asunnossa kaapelien avulla tai langattomasti (WLAN) (Tietoliikenteen ja tietotekniikan keskusliitto, FiCom ry 2016).

Langaton laajakaistayhteys mahdollistaa kiinteän laajakaistan käytön rajatun alueen sisällä. Kyseinen yhteys voidaan asettaa toimimaan esimerkiksi huoneistossa, joka pystytään toteuttamaan mm. WLAN-tekniikalla. Langaton laajakaistayhteys vaatii toimiakseen verkkosovittimen, joita ovat mm. modeemit. (Sanastokeskus TSK, 2012.) Alla kerrottuna hieman tarkemmin kiinteiden laajakaistayhteyksien tekniikoista.

2.2.2 Laajakaistayhteys tekniikoita

DSL-yhteys on lankapuhelinverkon kautta muodostettava kiinteä laajakaistayhteys, joka käyttää eri taajuusaluetta, kuin lankapuhelin. Kyseisen yhteystyyppin nopeus voi olla epäsymmetrinen tai symmetrinen. Symmetrinen tiedonsiirtonopeus tarkoittaa, että nopeus on sama ladattaessa tietoliikenneverkosta käyttäjälle päin tai käyttäjältä tietoliikenneverkkoon päin. Epäsymmetrinen tiedonsiirtonopeus puolestaan on erisuuruinen riippuen ladataanko dataa käyttäjälle päin vai käyttäjältä tietoliikenneverkkoon päin. (Sanastokeskus TSK, 2012.)

ADSL-yhteys on epäsymmetrinen DSL-yhteys, jossa käyttäjälle päin tuleva tiedonsiirto on nopeampi. ADSL-tekniikasta löytyy ADSL2+ versio, jossa yhteys voi saavuttaa maksimissaan 24M nopeuden. (Sanastokeskus TSK, 2012.) ADSL-

yhteys pystytään kytkemään lankapuhelinverkkoa hyödyntämällä (Tietoliikenteen ja tietotekniikan keskusliitto, FiCom ry 2016).

VDSL-yhteys on kiinteä laajakaistayhteys, joka on yleisesti epäsymmetrinen DSL-yhteys ja nopeampi, kuin ADSL-yhteys. Uudempi VDSL-tekniikka on VDSL2-tekniikka, jonka maksimi nopeus voi olla 100M. (Sanastokeskus TSK, 2012.) Myös VDSL-yhteys pystytään kytkemään lankapuhelinverkkoa hyödyntämällä (TeliaSonera, 2016).

Ethernet-yhteys on kiinteä laajakaistayhteys, joka yhdistetään yleiskaapeloinnilla käyttäen Ethernet-standardia. Ethernet ei vaadi käyttäjältä päätelaitetta toimiakseen vaan yhteyden saa muodostettua kytkemällä tietokoneen suoraan tietoliikenneverkkoon. (Sanastokeskus TSK, 2012.)

Valokuituyhteys on valokaapeilla muodostettava kiinteä laajakaistayhteys. Valokuituyhteyksissä yleisimmät toteutustavat ovat FTTH(fiber to the home), jossa valokuitu toimitetaan kodin huoneistojakamoon tai FTTB(Fiber to the building), jossa kuitu toimitetaan kiinteistön talojakamoon, josta laajakaistayhteydet jaetaan kiinteistössä asuville sisäverkkojohtoa pitkin. (Sanastokeskus TSK, 2012.)

Kaapelimodeemiyhteydessä tiedonsiirto tapahtuu kaapelitelevisioverkon avulla, jossa kaapelimodeemiyhteys käyttää eri taajuusaluetta, kuin kaapelitelevisiolähetykset (Sanastokeskus TSK, 2012).

2.3 Mobiililaajakaistayhteys

Mobiililaajakaistayhteys on matkaviestinverkon välityksellä muodostettava langaton laajakaistayhteys. Mobiililaajakaistayhteys pystytään muodostamaan mm. 4G- ja 3G verkoissa. Tiedonsiirronnopeuteen vaikuttaa samalla alueella olevien käyttäjien samanaikainen yhteysliikenteen käyttö. (Sanastokeskus TSK, 2012.) Alla kerrottuna hieman tarkemmin mobiililaajakaistayhteyksien tekniikoista.

2G-tekniikka on ensisijaisesti puheensiirtoon tarkoitettu toisen sukupolven matkaviestintekniikka. Kyseisessä tekniikassa puheensiirto on piirikytkentäistä eli päätelaitteiden välillä on pysyvä yhteys. Joissakin matkaviestinjärjestelmissä käy-

tetään piirikytkentäistä tiedonsiirtoa, kuten GSM:ssä. Mobiililaajakaistayhteyden pystyy muodostamaan käyttämällä pakettikytkentäistä EDGE-tekniikkaa. (Sanastokeskus TSK, 2012.)

3G-tekniikka on kolmannen sukupolven matkaviestintekniikka, joka mahdollistaa siirtää puhetta ja tietoa nopeasti. Kyseisellä tekniikalla pystytään muodostamaan mobiililaajakaistayhteys, jossa tiedonsiirto tapahtuu suurimmaksi osaksi pakettikytkentäisesti. Pakettikytkennässä toisin kuin piirikytkennässä päätelaitteiden välillä ei ole jatkuvaa yhteyttä. Pakettikytkennässä tiedonsiirto tapahtuu määrämuo-
toisissa jaksoissa. (Sanastokeskus TSK, 2012.)

4G-tekniikka on pakettikytkentäinen IP-yhteyksikäyttöön perustuva neljännen sukupolven matkaviestintekniikka. Kyseisellä tekniikalla pystytään muodostamaan mobiililaajakaistayhteys, joka mahdollistaa suurten puhe ja tietomäärien siirron. (Sanastokeskus TSK, 2012.)

2.4 Satelliittilaajakaistayhteys

Satelliittilaajakaistayhteys on langaton laajakaistayhteys, jonka muodostaminen tapahtuu satelliitin avulla. Kyseistä yhteyttä käytetään ns. harvaan asutuilla alueilla, joissa mobiililaajakaistayhteys tai kiinteä laajakaistayhteys eivät ole saatavilla. Satelliittilaajakaistayhteys pystyy muodostamaan yhteyden kaksisuuntaisesti satelliitin ja käyttäjän välillä. (Sanastokeskus TSK, 2012.)

3 OHJELMISTOTUOTANNON KÄYTÄNTÖJÄ

Ohjelmistotuotantoon kuuluu erilaisia käytäntöjä ja tässä kappaleessa käydään läpi ohjelmiston uudistamista, testausta ja ylläpitoa.

3.1 Ohjelmistotuotanto

Ohjelmistotuotanto (software engineering) tarkoittaa tietokoneohjelmistojen rakennuksessa käytettyjä työkaluja, tekniikoita, periaatteita ja menettelytapoja. Ohjelmistotuotannon päällimmäinen tarkoitus on täyttää asiakkaan odotukset tietokoneohjelman tuottamisesta siten, että aikataulu ja kehityskustannukset pystytään arvioimaan etukäteen mahdollisimman tarkasti. (Haikala & Mikkonen 2011, 11-12.)

Ohjelmistotuotantoon kuuluu suunnittelu, määrittely, toteutus, laadunvarmistus ja testaus. Näiden lisäksi erinäiset tuotannonprosessin alueet kuuluvat ohjelmistotuotannon piiriin. Näitä osa-alueita ovat projektihallinta, laatu järjestelmä, dokumentointi ja tuotteenhallinta. (Haikala & Mikkonen 2011, 12.)

Ohjelmistotuotannossa tarkoituksena on myös tunnistaa ja löytää ohjelmistojen tekemiseen soveltuvia toimintaperiaatteita, jotka ovat toimivia ja jotka mahdollistavat mm. paremman laadun (Kasurinen 2013, 22). Ohjelmistotuotanto organisoidaan usein projektiksi, jossa toteutetaan asiakkaan vaatimuksiin ja liiketoimintatavoitteisiin perustuva ohjelmisto (Haikala & Mikkonen 2011, 19).

3.2 Ohjelmiston uudistaminen

Ohjelmiston uudistamisen tavoitteena on vanhojen ohjelmien muuttaminen päivitetyiksi ohjelmiksi. Uudistaminen sisältää olemassa olevan järjestelmän analysoinnin, tutkimisen ja muutoksen päivitettyyn versioon. Tällä halutaan parantaa olemassa olevan järjestelmän laatua. (Harsu 2003, 177-178.) Ohjelmiston uudistamisesta puhuttaessa voidaan tarkoittaa joko ohjelmiston eri osien kehitystä tai koko ohjelmiston korvaamista. Uudistaminen itsessään on paljon suurempi prosessi, kuin ohjelman normaali ylläpito, joten sen vuoksi ohjelmiston uudistamiset toteutetaan projektina. (Harsu 2003, 165.)

Uudistamista ei siis tule sekoittaa ylläpitoon, sillä ohjelman ylläpidolla tarkoitetaan ohjelmistoon toteutettavaa pientä muutosta, kun taas ohjelmiston uudistaminen on laajempi ja suunnitelmallisempi toimenpide, jonka tarkoituksena on parantaa ohjelman rakennetta laaja-alaisemmin kuin ylläpidossa. (Harsu 2003, 178.) Toisin sanoen uudistaminen on ennalta ehkäisevää ylläpitoa (Harsu 2003, 24).

Ohjelmiston uudistamista voidaan tarkastella kuudesta eri näkökulmasta, joita ovat: tekninen, liikkeenjohdollinen, evoluutionäkökulma, ylläpidollinen, ohjelmallinen ja järjestelmänäkökulma (Harsu 2003, 165).

3.3 Ohjelmiston testaus

Ohjelmiston testaus on ohjelmistotuotantoon kuuluva osa-alue, jossa testaaja käy läpi sen mitä on kehitetty. Testausvaiheessa tarkastetaan vastaako tuote sitä mitä on tehty ja tarkoituksena on tunnistaa toimiiko ohjelmisto, kuten on suunniteltu. (Kasurinen 2013, 10.) Ohjelmistojen testaus on suunnitelmallista virheiden etsimistä, jossa suoritetaan ohjelmaa tai sen osaa. Testauksen tarkoituksena on tuoda esille mahdolliset ohjelmassa esiintyvät virheet, mutta koko ohjelman virheettömyyttä ei testauksella pystytä todistamaan. (Haikala & Mikkonen 2011, 205.)

Testauksen työvaiheita ovat suunnittelu, johon sisältyvät testitapaukset ja testaus-suunnitelma, testiympäristön kehittäminen, testin suorittamiset ja lopputulosten tarkastaminen (Haikala & Mikkonen 2011, 205). Testauksen päätavoite on varmistaa, että luotu ohjelmisto on toteutettu suunnitelman ja asiakasvaatimusten mukaisesti. Testausvaihe päättyy, kun ohjelmistossa ei esiinny merkittäviä virheitä tai, että ohjelmisto on niin toimintakuntoinen, että se toteuttaa kaikki sille asetetut vaatimukset. Testausvaiheen saavutettua määränpänsä ohjelmisto on valmis käyttöönottettavaksi asiakasympäristössä ja tuote siirtyy ylläpitoon. (Kasurinen 2013, 13.)

ISTQB-testaussertifikaatissa on määritetty testaukselle seitsemän periaatetta, joilla ohjelmistoille tehtävät testaukset voidaan perustella ja määritellä. Periaatteet on määritelty seuraavasti:

1. Testauksella osoitetaan virheiden olemassaolo. Testauksessa on tarkoituksena näyttää, että testauksen alla olevassa ohjelmistossa on virheitä, mutta tarkoituksena samalla vähentää todennäköisyyttä virheiden olemassaololle. Vaikka testauksessa ei virheitä löydetä, ei se tarkoita että tuotteessa ei olisi virheitä.
2. Täydellistä testausta ei voida tehdä. Testauksen tulisi perustua testaus-työn priorisointiin sekä kartoittamaan riskejä.
3. Testauksen aloitus tulee tehdä mahdollisimman aikaisin. Testauksessa tulee ottaa huomioon myös vaatimusmääritelmät ja projektissa tehdyt muut suunnitelmat.
4. Testaus tulee painottaa sellaisiin osiin ja moduuleihin, joissa odotetaan tai tiedetään esiintyvän eniten vikoja.
5. Ennestään testatut testitapaukset tulisi muistaa testata uudelleen. Ilman uudelleen testausta päädytään tilanteeseen, jossa jo aiemmin testatuissa testitapauksissa esiin nostetut virheet on korjattu, mutta korjauksen jälkeen voi nousta esiin uusia virheitä. Tämän vuoksi kannattaa testitapauksiin muistaa tarvittaessa kirjata päivitykset, kehitykset ja lisäykset.
6. Testausta joudutaan tekemään eri tavalla riippuen tilanteesta ja projektista. Tämän takia prosessien ja testauksen hallinnointimallissa tulee olla joustoa.
7. Virheiden korjaaminen ja löytäminen ei auta, jos järjestelmä on suunniteltu väärin eikä toteuta sille asetettuja vaatimuksia. (Kasurinen 2013, 48-49.)

Yhteenvedona voidaankin todeta, että testaus ei ole pelkkää toimintojen toteamista ja ohjelman käyttöä. Testauksen tarkoituksena on erilaisten toimintojen kokeilun ja rakenteen arvioimisen lisäksi analysoiminen, raportointi, tiedonhallitseminen sekä suunnittelu. Testauksessa on erilaisia työvaiheita ja niissä keskitytään erilaisiin asioihin, joissa käytetään erilaisia testausmenetelmiä ja joiden avulla saadaan erilaisia lähestymistapoja mahdollisia ohjelmistovirheitä etsiessä. (Kasurinen 2013, 49.)

3.4 Ohjelmiston ylläpito

Ylläpitovaiheessa ohjelmaan tehdään tarpeen vaatiessa korjauksia, jos löydetään ongelmallisia vikoja. Ylläpidossa ohjelman korjausten lisäksi voidaan lisätä uusia toiminnallisuuksia, jos lisäys on perusteltavissa. (Kasurinen 2013, 13.)

Ylläpito on vaihe ohjelmistotuotannossa, joka alkaa ohjelmiston käyttöönotosta ja päättyy, kun ohjelmisto poistetaan käytöstä (Harsu 2003, 18). Ylläpidon tarkoituksena on toteuttaa uuden ympäristön mukauttamista, virheiden korjaamista, ylläpidettävyyden parantamista ja uusien toimintojen lisäyksiä. Ylläpidon aikana tehtävissä muutoksissa tulee aina huomioida vaikutukset, joita tehdyt muutokset aiheuttavat. Pienetkin muutokset voivat vaatia muutoksia useampaan osaan ohjelmistossa. Toisaalta toisen asian korjaaminen ohjelmistossa voi aiheuttaa uuden virheen syntymisen toisessa osassa ohjelmaa. (Harsu 2003, 77.)

Ylläpito voidaan jakaa täydentävään, korjaavaan ja adaptiiviseen ylläpitoon. Täydentävässä ylläpidossa parannetaan ohjelmaa lisäämällä tai muuttamalla sen toiminnallisuuksia ja korjaavassa ylläpidossa puolestaan korjataan ohjelmassa esiintyviä virheitä ja adaptiivisessa ylläpidossa ohjelmaa muutetaan esim. ympäristön vaatimusten muutosten vuoksi. (Haikala & Märijärvi 2004, 41.)

Ohjelmien ylläpitoa ei tehdä ainoastaan virheiden korjaamiseksi vaan ohjelmia täytyy muuttaa, jotta ne pystyvät toimimaan uudistettujen käyttöjärjestelmien ja laitteiden kanssa. Ohjelmistoja tulee myös muuttaa tarvittaessa käyttäjien tarpeiden mukaan. (Harsu 2003, 18.)

Ohjelmien ylläpidolle on useita syitä, jotka jaotellaan seuraavasti:

1. Testausvaiheessa ei aina paljastu kaikki ohjelmassa olevat virheet vaan ohjelmiston käyttäjät löytävät niitä ohjelmaa käyttäessään. Tällöin puhutaan korjaavasta ylläpidosta (corrective maintenance).
2. Laitteistoja muutetaan hyvinkin nopealla tahdilla, mutta ohjelmistoja ei korvata samalla tahdilla, joten ohjelmistoilta vaaditaan mukautumista käyttöjärjestelmiin ja uusiin laitteisiin. Tällöin puhutaan mukautuvasta ylläpidosta (adaptive maintenance).

3. Käyttäjät voivat toivoa ohjelmaan muutoksia tai uusia ominaisuuksia. Tällöin puhutaan täydellistä ylläpidosta(perfective maintenance).
4. Ohjelmia joudutaan muuttamaan ja parantamaan, jotta tulevat muutostoi-
met ja ylläpito ovat helpommin tehtävissä. Tällöin puhutaan ehkäisevästä
ylläpidosta(preventive maintenance). (Harsu 2003, 18-19.)

4 UML

Tässä kappaleessa kerrotaan UML:stä ja siihen liittyvistä mallinnuskaavioista. Kaavioita on 14 erilaista ja tässä opinnäytetyössä syvennyttään tarkemmin tila-, aktiviteetti-, sekvenssi- ja käyttötapauskaavioon. Näiden lisäksi käydään läpi käyttäjätarinat sekä käyttötapaukset.

4.1 UML ohjelmistotuotannossa

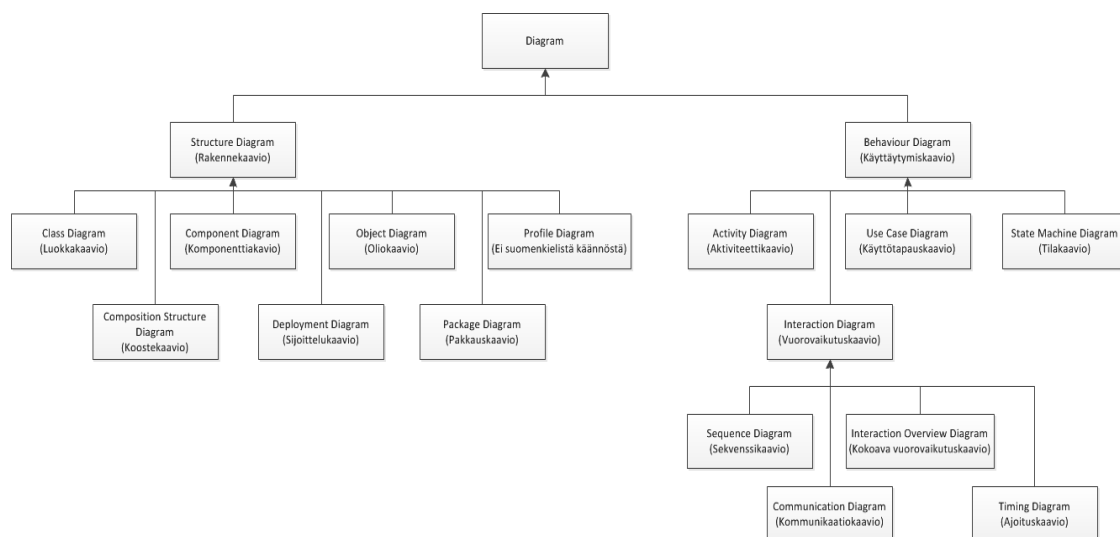
UML(Unified Modeling Language) on graafinen olio-ohjelmointiin tarkoitettu mallinnuskieli. Kyseessä on yleiskäyttöinen mallintamiskieli, mikä tarkoittaa, että sen käyttöä ei ole rajoitettu tiettyyn sovellusalueeseen. UML-mallinnuksen tuloksena on graafinen malli, joka tarjoaa erilaisten kaavioiden avulla erilaisia näkökulmia mallinnetusta järjestelmästä. UML ei ole sidonnainen mihinkään ohjelmointikieleen, sovelluksen käyttöympäristöön tai kehitystyökaluun eikä se tarjoa myöskään ohjelmistokehitysprosessia. UML:ää voidaan myös käyttää käynnissä olevien ohjelmistokehitysprosessien aikana. (Seidl, Scholz, Huemer & Kappel 2015, 14.)

UML-kaaviot kuuluvat vahvasti ohjelmistotuotantoon ja niiden avulla pystytään helposti kuvaamaan kokonaisuutta, jota ollaan rakentamassa sekä helpottamaan kommunikointia eri osapuolten välillä. Lähtökohta UML-kielessä on tarjota ohjelmiston suunnitteluun erilaisia kaavioita ja näkymiä. Näiden kaavioiden avulla eri käyttäjäryhmät pystyvät esittelemään suunnitelmiaan ja vaihtamaan ajatuksiinsa. (Kasurinen 2013, 29.)

UML:ssä ajatuksena on, että pystytään tekemään kuvauksia jopa ohjelmakoodin tasolle asti, mutta ymmärrys säilyy myös sellaisille henkilöille, jotka eivät varsinaisesti ohjelmointityön kanssa ole tekemisissä. Tämän lisäksi mallinnuskielen tarkoituksena on pystyä yhdistämään vaatimusmäärittelyssä esitetyt ohjelmiston fyysinen rakenne, vaatimukset sekä esimerkiksi järjestelmän toimintalogiikka, jotta järjestelmää luovat henkilöt pystyvät ymmärtämään ja tarkastamaan kokonaisuutena, miten ohjelmisto on rakennettu. (Kasurinen 2013, 29.)

4.2 UML-kaaviot

UML:ssä mallit esitetään graafisesti diagrammeilla. On olemassa diagrammeja, joilla kuvataan mitä toiminnallisuutta kukin käyttäjä käyttää. Näitä diagrammeja kutsutaan käyttäytymiskaavioiksi (behaviour diagrams). On olemassa myös diagrammeja, joilla esitetään järjestelmän rakenne. Näitä kutsutaan rakennekaavioiksi (structure diagrams). Kehitettynä on myös diagrammeja, joilla voidaan esittää tuetut ja kielletyt prosessit. (Seidl ym. 2015, 15.)



Kuvio 1 UML-kaaviot (Seidl ym. 2015, 15.)

Kuviosta 1 näkee, miten UML-kaaviot on kategorisoitu. Structure diagramin (rakennekaavio) alla on 7 erilaista kaaviota, jotka ovat class diagram (luokkakaavio), component diagram (komponenttikaavio), object diagram (oliokaavio), composition structure diagram (koostekaavio), deployment diagram (sijoittelukaavio), package diagram (pakkauskaavio) ja profile diagram, jolle ei ole suomenkielistä nimeä kehitettynä.

Behaviour diagramin (käyttäytymiskaavio) alla puolestaan on 4 erilaista diagrammia, jotka ovat activity diagram (aktiviteettikaavio), use case diagram (käyt-

tötapauskaavio), state machine diagram (tilakaavio) ja interaction diagram (vuorovaikutuskaavio). Interaction diagram koostuu 4 erilaisesta diagrammista, jotka ovat sequence diagram (sekvenssikaavio), interaction overview diagram (kokoava vuorovaikutuskaavio), communication diagram (kommunikaatiokaavio) ja timing diagram (ajoituskaavio).

Tässä opinnäytetyössä UML-kaavioista käydään tarkemmin läpi aktiviteettikaavio, tilakaavio, käyttötapauskaavio ja sekvenssikaavio.

4.3 Käyttäjätarinat

Käyttäjätarina(user story) on ketterässä ohjelmistokehityksessä käytettävä dokumentointi tapa, josta on tarkoituksena selvittää mm. aktori eli kuka tekee, mitä tuliaan tekemään ja mitä lisäarvoa se tuottaa (Haikala & Mikkonen 2011, 83-84).

Käyttäjätarinat on hyvä tapa aloittaa ohjelmistoprojekti. Käyttäjätarinoita kirjataan mm. asiakasedustajien ja tulevien käyttäjien avulla. Käyttäjätarinoihin voidaan ottaa myös mukaan tulevat järjestelmän kehittäjät. Hyvät käyttäjätarinat pitävät seuraavanlaisia piirteitä sisällään:

- Riippumattomuus: Käyttäjätarinoiden tulee olla riippumattomia toisistaan, jotta työmäärän arviointi toteutuksen ja priorisoinnin osalta ei osoittaudu mahdottomaksi.
- Arvokkuus: Arvokkuudella tarkoitetaan lisäarvoa, jota käyttäjätarinan tulee käyttäjille luoda.
- Neuvoteltavuus: Käyttäjätarinoiden tulee olla myös neuvoteltavissa eli tarpeen vaatiessa niitä tulee pystyä muuttamaan.
- Pienuus: Pienet käyttäjätarinat tuovat yksinkertaisuutta ja ymmärrettävyyttä.
- Arvioitavuus: Käyttäjätarinoiden tulee olla myös arvioitavia, koska ohjelmiston suunnittelu tehdään käyttäjätarinoiden pohjalta.
- Testattavuus: Tarinat pitää olla myös sellaisia, jotta ne voidaan testata. (Sininen meteoriitti, 2013.)

Käyttäjätarinoiden ei tule olla täydellisiä määrittelyitä, vaan muistutus asioista, joista tulee keskustella. Käyttäjätarinoiden tekeminenkään ei ole aina hyvä ratkaisu, koska suurimmissa projekteissa tarinoita voi olla niin suuri määrä, että kaikkien läpikäynti ja käsittely voi olla mahdotonta. Käyttäjätarinoiden yksi perimmäisistä tarkoituksista on parantaa suullista viestintää. (Sininen meteoriitti, 2013.)

4.4 Käyttötapaukset

Käyttötapausten (use cases) tarkoituksena on kartoittaa asiakasvaatimukset ja kuvata niiden perusteella ohjelmistovaatimukset. Käyttötapaus alkaa käyttäjäroolista ja päättyy, kun tahdottu toiminnallisuus on käsitelty onnistuneesti loppuun tai se on tuottanut lisäarvoa käyttäjälleen. Hyvän käyttötapausten ominaisuuksia ovat mm. ymmärrettävyys, testattavuus, sopiva tarkkuus ja koko. Ohjelmiston käyttötapaukset ja käyttäjäroolit kuvataan käyttötapaustaaviossa. (Haikala & Märijärvi 2004, 158-159.)

Käyttötapausten avulla kuvataan järjestelmässä toimivien käyttäjien toiminnallisuutta ja heidän tekemiään tapahtumaketjuja. Tavallisesti kuvaukset tehdään tekstuaalisesti, mutta apuna voidaan käyttää esimerkiksi tapahtumasekvenssikaaviota. Jokaisella käyttötapaustella on vähintään yksi käyttäjärooli(actor), jonka tehtävänä on suorittaa määritetty käyttötapaus. (Haikala & Märijärvi 2004, 158.)

Käyttötapausten monikäyttöisyys on tehnyt niistä yleistävän toiminnon järjestelmien luonnissa ja kehityksessä. Käyttötapausten tarkoituksena on saada liitettyä asiakasvaatimukset järjestelmän toimintoihin. Yksi käyttötapaus voi sisältää useamman ohjelmaan toteutettavan toiminnon. Käyttötapaustissa uudelleenkäytettävyys on myös hyvin yleistä, kun yleensä samat toiminnot alkavat toistumaan useissa eri käyttötapaustissa. Asiakasvaatimusten tarkentuessa varmistutaan siitä, että toimittaja ja asiakas ovat yhtä mieltä vaatimuksista. (Haikala & Mikkonen 2011, 83.)

UML:ssä käyttötapausten kuvauksia ei ole standardoitu, mutta täytyy muistaa, että kuvaus on tärkein osa käyttötapausta, jossa esitetään mitä käyttötapaustissa tulisi tapahtua. Käyttötapaustissa kuvataan normaalisti toiminteen onnistuminen,

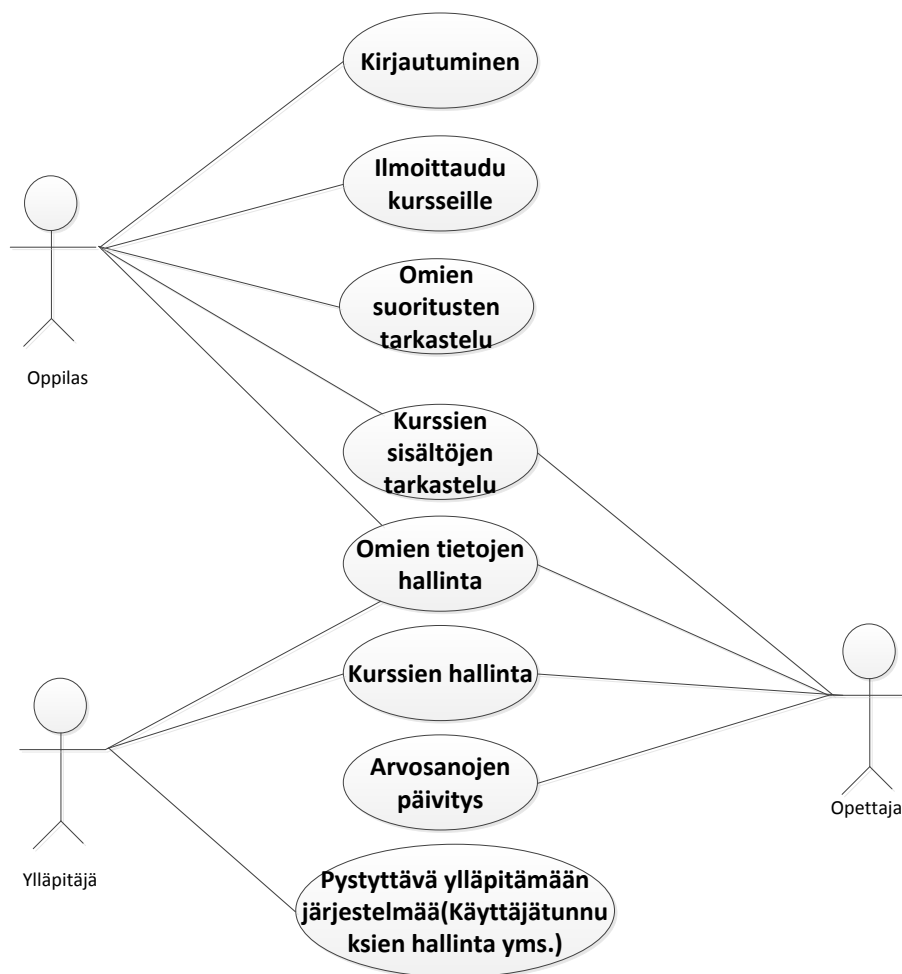
kun taas ongelmatapaukset ilmoitetaan poikkeuksissa. Tällaiset poikkeustapaukset kirjataan itse käyttötapaukseen tai muuhun käyttötapaukseen liittyvään tekstiin. Kannattaa myös huomioida, että ei-toiminnallisia vaatimuksia ei UML:ssä normaalisti oteta huomioon, mutta käyttötapausten yhteyteen voidaan tällaiset tapaukset kirjata. (Haikala & Mikkonen 2011, 79.)

4.5 Käyttötapauskaavio

Käyttötapauskaaviot (use case diagrams) kuuluvat UML:n kaaviotekniikkoihin. Käyttötapauskaavioilla määritellään ohjelmaan tulevat päätoiminnallisuudet ja tarkoituksena on kuvata työvaiheittain, miten ohjelmaa tullaan käyttämään. (Kasurinen 2013, 30.) Käyttötapauskaaviot sisältävät aktorit ja käyttötapaukset, jotka yhdistetään toisiinsa viivoilla. Käyttötapauskaavioissa kuvatut aktorit ovat järjestelmän käyttäjiä. (Haikala & Mikkonen 2011, 78.) Käyttötapauskaavioilla on tarkoitus pystyä ilmaisemaan yleisellä tasolla, miten käyttäjät kytkeytyvät merkittävihin käyttötapauksiin ja mitkä käyttötapaukset liittyvät toisiinsa (Kasurinen 2013, 31).

Käyttötapauksista tehdään myös tarkemmat sanalliset kuvaukset, jossa kerrotaan mitä käyttötapauksessa tapahtuu. Sanallisessa kuvauksessa käydään vaihe vaiheelta läpi, mitä käyttäjän tulee tehdä, että käyttötapaus toteutuu sekä mitä aiemmin on pitänyt tapahtua. (Kasurinen 2013, 31.)

Käyttötapauskaavioiden avulla pystytään kuvaamaan mahdolliset käyttötapaukset. Käyttötapauskaavio ilmaisee, mitä järjestelmän pitäisi tehdä, mutta se ei ota kantaa toteuttamisen yksityiskohtiin. Käyttötapauskaaviolla mallinnetaan myös, mitä toiminnetta järjestelmässä kukin käyttäjä (actor) käyttää. (Seidl, ym. 2015, 23) Kuviossa 2 on kuvattuna esimerkkitapaus, käyttötapauskaaviossa esiintyvistä aktoreista, käyttötapauksista ja miten ne yhdistetään käyttötapauskaaviossa toisiinsa.



Kuvio 2 Käyttötapauskaavio (Haikala & Mikkonen 2011, 77)

Käyttötapauskaavioiden tehtävä on helpottaa ohjelmiston määrittelyvaiheesta siirtymistä ohjelmiston toteutus vaiheeseen. Käyttötapauskaavion tarkoituksena on auttaa toimintojen pilkkomisessa osajärjestelmiin. Käyttötapauskaaviota voidaan käyttää hyödyksi ohjelmistokehityksen organisoinnissa sekä testitapausten suunnittelu vaiheessa. Käyttötapauskaavioilla pystytään myös helposti rajaamaan ja hahmottamaan järjestelmää sekä ne auttavat ja helpottavat tunnistamaan järjestelmässä eri sidosryhmät. Käyttötapauskaavioita pystytään myös uudelleen käyttämään, jos samoja toiminnallisuuksia tarvitsee käyttää useammassa käyttötapauskaavioissa. (Haikala & Mikkonen 2011, 83.)

4.6 Sanalliset käyttötapaukset

Käyttötapauksiin liittyy aina sanalliset käyttötapaukset, joissa kuvataan sanallisesti, mitä käyttötapauksessa tapahtuu. Sanallisissa kuvauksissa käydään vaihe vaiheelta läpi, miten toimijan pitää edetä, että käyttötapauksen eri ehdot ja määränpää toteutuvat. (Kasurinen 2013, 31.)

Sanallisten käyttötapauksien luomisen yhteydessä on hyvä muistaa käyttää lyhyitä lauseita, selkeää ja yksinkertaista kieltä, jolloin sanallisen käyttötapauksen tarina etenee vaiheittain. Sanalliset käyttötapaukset myös helpottavat tilaajaa ymmärtämään järjestelmän toiminnallisuutta, jolloin tilaajan on helppo antaa omia ehdotuksia ja mielipiteitä järjestelmän kehityksen suunnasta. Sanalliset käyttötapaukset tukevat myös testausvaihetta, jolloin sanallisten käyttötapauksien avulla pystytään seuraamaan, että järjestelmä toimii, kuten on suunniteltu. Sanallisten käyttötapauksien pituudeksi suositetaan yhtä A4-arkkia. (Haikala & Mikkonen 2011, 80-81.)

Käyttötapauksia kirjattaessa on muistettava asiakkaan määrittämät tarpeet ja tämän vuoksi käyttötapauksen tulee sisältää tarkka kuvaus siitä, miten käyttäjän tarve saadaan täytettyä. Kirjattaessa käyttötapauksia kannattaa huomioida ainakin seuraavat asiat:

- Kuvaukset ovat sopivia eli ei mennä liian tarkalle tasolle.
- Käytettävä selkeää ja yksinkertaista kieltä vaiheistamaan käyttötapauksen tarina.
- Kaikkia asiakasvaatimuksia ja käyttötilanteita ei voida kirjata käyttötapauksiksi.
- Kirjattava selkeästi aloitustilanne ja lopetustilanne.
- Yksi käyttötapaus sisältää useamman kuin yhden ohjelman toiminnon.
- Käyttötapauksista ei saa tulla liian pitkiä. (Haikala & Mikkonen 2011, 81.)

Yksinkertaisuutensa ansiosta käyttötapauksien tulkinta on helppoa, mutta jotta asiakas pystyy niitä tulkitsemaan, tulee niiden olla kirjoitettuna asiakkaan ymmärtämällä terminologialla ja kielellä (Haikala & Mikkonen 2011, 79). Sanallisissa käyttötapauksissa on hyvä välttää kirjaamasta liian yksityiskohtaisesti käyttöliit-

tymän toiminnallisuuden mukaisia käyttötapauksia, koska muuten toiminnallisuuden toteuttajalle voi tulla vaikeuksia toteuttaa toiminnallisuuksia ja sanallisista käyttötapauksista tulee liian monimutkaisia (Haikala & Mikkonen 2011, 82).

Nimi: Kurssille ilmoittautuminen

Versio: 0.1

Luoja: Pasi Ojala

Toimija/Actor: Opiskelija

Tavoite: Saada kurssipaikka varattua

Alkuehdot: Opiskelija, opintojakso ja kurssi on syötettynä järjestelmään. Kurssi on avattu järjestelmässä. Opiskelijalla on oikeus kirjautua kurssivarausjärjestelmään. Opiskelija on kirjautunut järjestelmään.

Kuvaus: Opiskelija valitsee kurssivarausjärjestelmästä ilmoittaudu vaihtoehdon. Opiskelija valitsee kurssin/kurssit, joihin haluaa ilmoittautua ja klikkaa Vahvista painiketta.

Poikkeus tapaukset: Opiskelijalla ei oikeutta järjestelmään. Ilmoittautumisaika on sulkeutunut. Kurssi on täynnä.

Lopputulos: Opiskelija on onnistuneesti rekisteröitynyt valituille kursseille.

Muita vaatimuksia: -

Kuvio 3 Sanallinen käyttötapaus (Haikala & Mikkonen 2011, 80)

4.7 Tilakaavio

Tilakaaviot ovat mallinnusvälineitä, joilla mallinnetaan mm. tietokoneprotokollia, liiketoimintaprosesseja, elektromekaanisten laitteiden toimintaa ja ohjelmointikielten jäsentämistä. Tilakaaviot ovat ainoita UML-kaavioita, joilla voidaan tehdä hahmottaessa erittäin tarkkoja mallinnuksia. Normaalisti tilakaavioita on käytetty oliomallinnuksen yhteydessä mallintamaan olioiden elinkaarta, mutta nykyisin tilakaavioita käytetään monella muullakin tavalla. (Haikala & Mikkonen 2011, 105.)

Tilakaaviossa kuvataan aina tapahtumaketju alusta loppuun määrittelyllä, jota kutsutaan tilakoneeksi. Tilakaavion keskeisin osa on se, että tiedetään koko ajan tapahtuma, joka järjestelmässä on meneillään. Tilakaaviossa voidaan myös merkitä, jos halutaan järjestelmän odottavan tiettyjen ehtojen tai toiminnallisuuksien täyttymistä, ennen kuin voidaan siirtyä seuraavaan vaiheeseen. Näiden lisäksi tilakaavioon voidaan määritellä eri tilojen välillä ehtoja, jotka kertovat miten eri tilojen välillä voidaan siirtyä. Tilakaavioilla pystytään kuvaamaan kaikkia mahdollisia

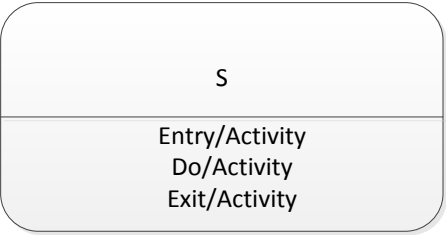
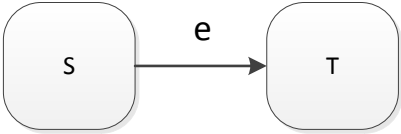



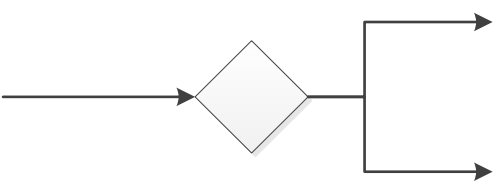
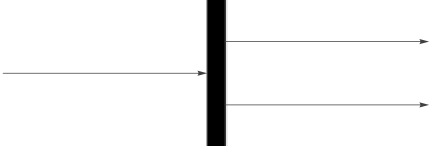
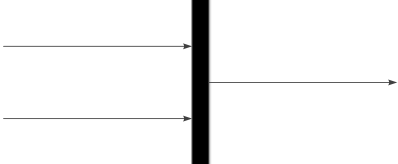
tiloja, joihin olio voi siirtyä sekä miten kyseiseen oloon erinäiset kohdistuvat tapahtumat muuttavat sen tilaa. (Kasurinen 2013, 38.)

Tilakaavioissa tärkeitä käsitteitä ovat tilat ja tilasiirtymät ja näihin liittyvät mahdolliset toiminnot. Tilakone on ”odottaa” tilassa, joka odottaa herätettä, joka aiheuttaa tilasiirtymän toiseen tilaan tai takaisin tilaan, josta lähti. Tilasiirtymiin voidaan asettaa ehtoja, joilla tilasiirtymä estetään tai sallitaan. Tilakaavioita on olemassa kahta erilaista, joista toinen on Mooren ja toinen Mealyn tila-automaatti. Mooren automaatti tilakoneessa toiminnot suoritetaan, kun automaatti on jossain tilassa, kun taas Mealyn automaatissa toiminnot suoritetaan tilasiirtymien aikana. (Haikala & Märijärvi 2004, 138.)

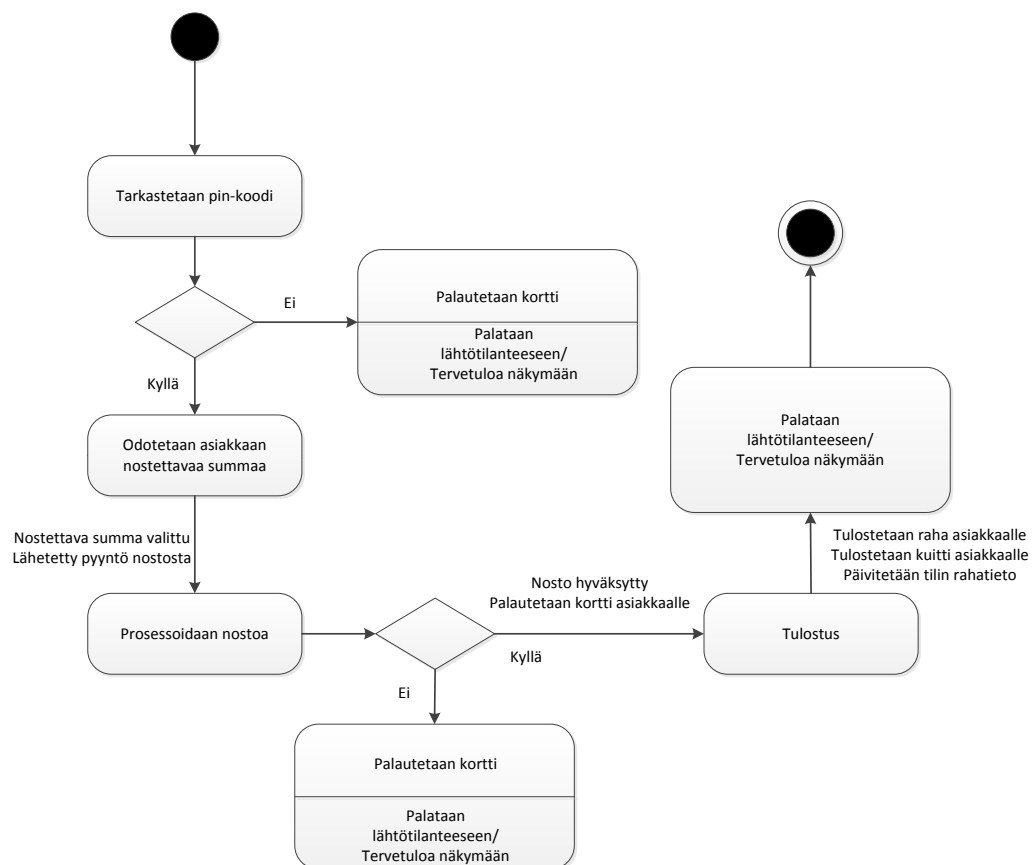
Jotta tilakone voidaan määritellä, täytyy sille luetella sen tilat ja jokaiselle tilalle on määritettävä tilan aikana tehtävät toiminnallisuudet. Tämän lisäksi on määriteltävä erilaiset tilasiirtymät, jotka tilasta voi tapahtua ja niihin liittyvät toiminnot. Lopuksi tilakoneelle on määritettävä aina alkutila ja lopputila tai mahdollisesti joukko lopputiloja. UML:n tilakoneissa voidaan määritellä tarpeen vaatiessa suoritettava toiminto, kun saavutaan tiettyyn tilaan, kun ollaan tiettyssä tilassa sekä kun poistutaan tietystä tilasta. (Haikala & Märijärvi 2004, 138-139.)

Tilakoneessa voi esiintyä myös transienttitiloja, joiden tarkoituksena on yksinkertaistaa tilakaavioita. Transienttitilasta lähteviin tilasiirtymiin ei aseteta signaaleja, jotka laukaisevat tilasiirtymän. Saavuttaessa transienttitilaan tilasiirtymä ei jää odottamaan ns. herätettä vaan siirtyy heti seuraavaan tilaan. (Haikala & Märijärvi 2004, 139.)

Kuviossa 4 on esitelty tilakaaviossa käytettäviä symboleja. Kuviossa 5 on mallinnettu tilakaaviolla rahan nostaminen raha-automaatista.

	State
	Transition
	Initial state
	Final state
	Terminate mode
	Decision mode
	Parallelization node
	Synchronization node

Kuvio 4 Tilakaavion symboleja (Seidl ym. 2015, 106)



Kuvio 5 Tilakaavio (Wikispaces, 2016)

4.8 Aktiviteettikaavio

Aktiviteettikaavioilla keskitytään mallintamaan järjestelmän näkökulmasta prosessuaalista käsittelyä. Niillä myös täsmennetään järjestelmän ohjaus ja datan siirtyminen eri vaiheiden välillä. Aktiviteettikaaviot myös perustuvat vakiintuneisiin konsepteihin kuvattaessa samanaikaisten prosessien välistä kommunikointia. Näiden kaavioiden ominaispiirteeseen kuuluu mallintaminen sekä oliosuuntautuneisesti ja ei-oliosuuntautuneisesti. Tämä mahdollistaa aktiviteettien määrittelyn riippumatta olioista, mikä tarkoittaa että voidaan mallintaa funktioiden ohjelmakirjastot yhtä hyvin kuin liiketoimintaprosessit (Seidl, ym. 2015, 141.)

Aktiviteettikaavioita käytetään määrittely- ja suunnitteluvaiheessa. Määrittelyvaiheessa kaavioilla pystytään kuvaamaan yksittäisten käyttötapausten toiminnan


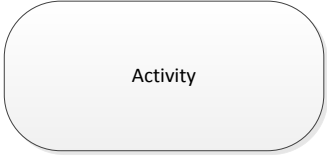


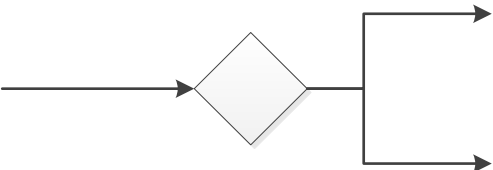
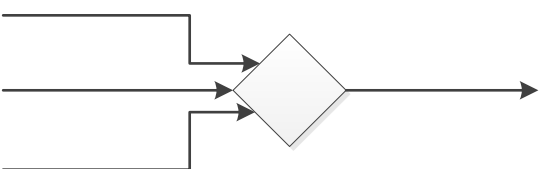
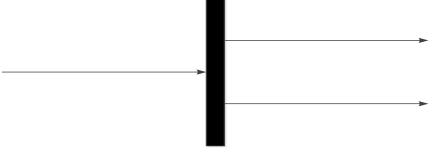
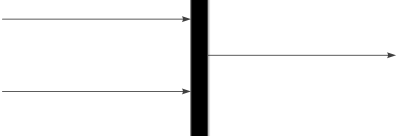
etenemistä tai useimpien käyttötapausten välisiä toimintoja. Suunnitteluvaiheessa puolestaan näillä kaavioilla voidaan kuvata mm. metodien toiminnan etenemistä. (Enden 2000, 7) Aktiviteettikaaviossa esitellään normaalisti tietyn tapahtuman erikoistapaukset, joita kutsutaan transienttitiloiksi. Kyseisessä tilassa ei jäädä odottamaan tapahtuman päättymistä vaan tahdottu toiminne jatkaa seuraavaan vaiheeseen. (Haikala & Mikkonen 2011, 106.)

Aktiviteettikaaviossa alkutiloja voi olla vain yksi. Alkutila on ns. heräte, joka käynnistää toiminnallisuuden. Toiminnot kuvataan suorakaiteilla, joiden kulmat on pyöristetty. Toiminnossa aktiviteetti kuvataan suorakaiteen sisään kirjoitettavalla tekstillä. Toiminnon taso voi olla alkeistoiminto, yksittäinen toiminto tai toimintokokonaisuus (aliprosessi). Alkeistoiminnoksi kutsutaan toimintoa, joka alkaessaan suoritetaan aina loppuun tai sitten toimintoa ei suoriteta ollenkaan. (Sahi, 2010.)

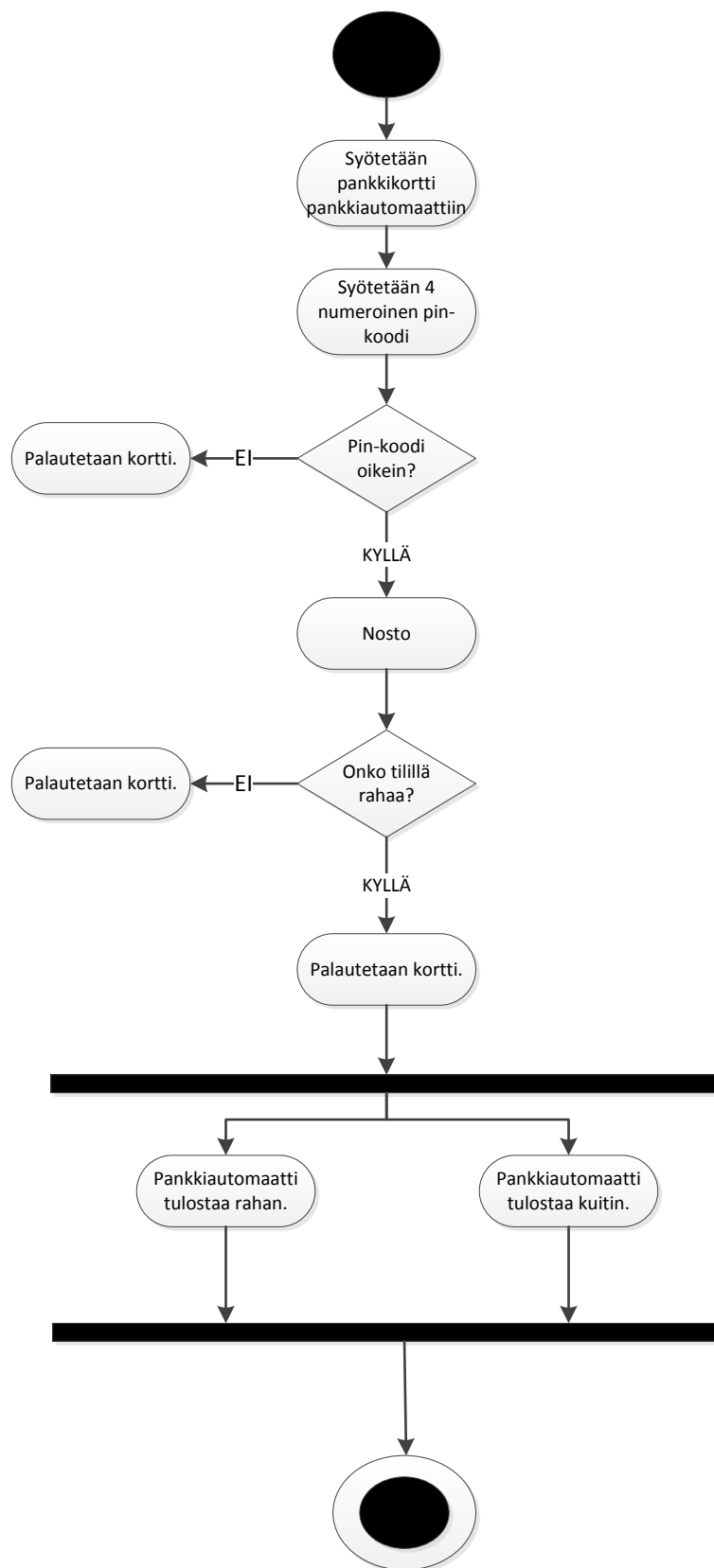
Nuolilla kuvataan siirtymät ja siirtymien aikana voi esiintyä päätöksentekoa vaativia tapahtumia, joita kuvataan kärjellään olevilla neliöillä, josta lähtee suoritusehdot. Päätöksentekomerkillä voi olla yksi tai useampi saapuva siirtymä, jonka lisäksi sillä on kaksi tai useampi poistuva siirtymä. Näistä poistuvista siirtymistä ainakin yksi toteutuu. (Sahi, 2010.)

Siirtymille tyypillistä on myös haarautuminen ns. rinnakkaisiksi suoritussäikeiksi, jonka jälkeen ne yhtyvät synkronoidusti. Tällaista tilannetta kuvataan vaakasuoraisella mustalla palkilla, josta säikeet lähtevät. Samat säikeet yhtyvät samantyyppisessä mustassa palkissa, jota kutsutaan myös synkronointipalkiksi. (Sahi, 2010.)

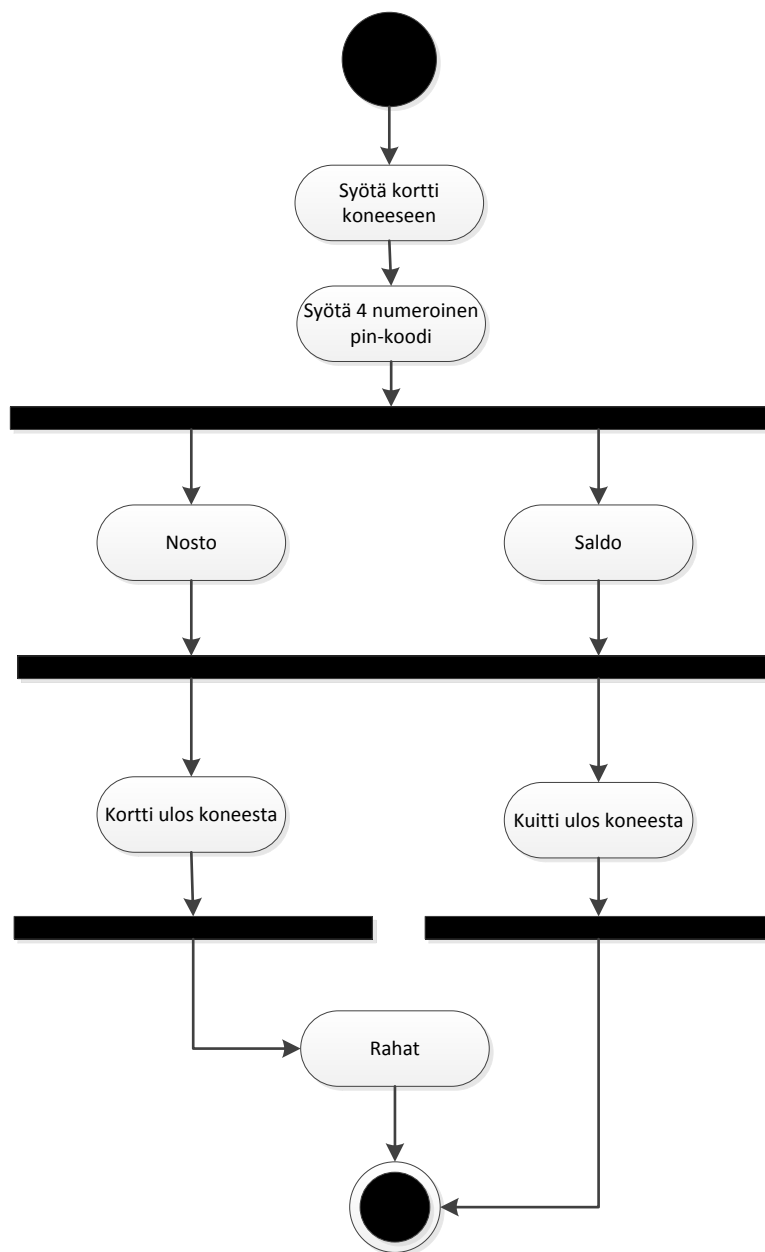
Kuviossa 6 on esitelty aktiviteettikaaviossa käytettäviä symboleja. Kuviossa 7 on mallinnettu aktiviteettikaaviolla rahan nostaminen raha-automaatista. Kuviossa 8 on mallinnettu toisella tavalla rahan nostaminen raha-automaatista aktiviteettikaaviota käyttämällä.

	Action node
	Activity node
	Initial node
	Activity final node
	Decision node
	Merge node
	Parallelization node
	Synchronization node

Kuvio 6 Aktiviteettikaavion symboleja (Seidl ym. 2015, 165)



Kuvio 7 Aktiviteettikaavio 1



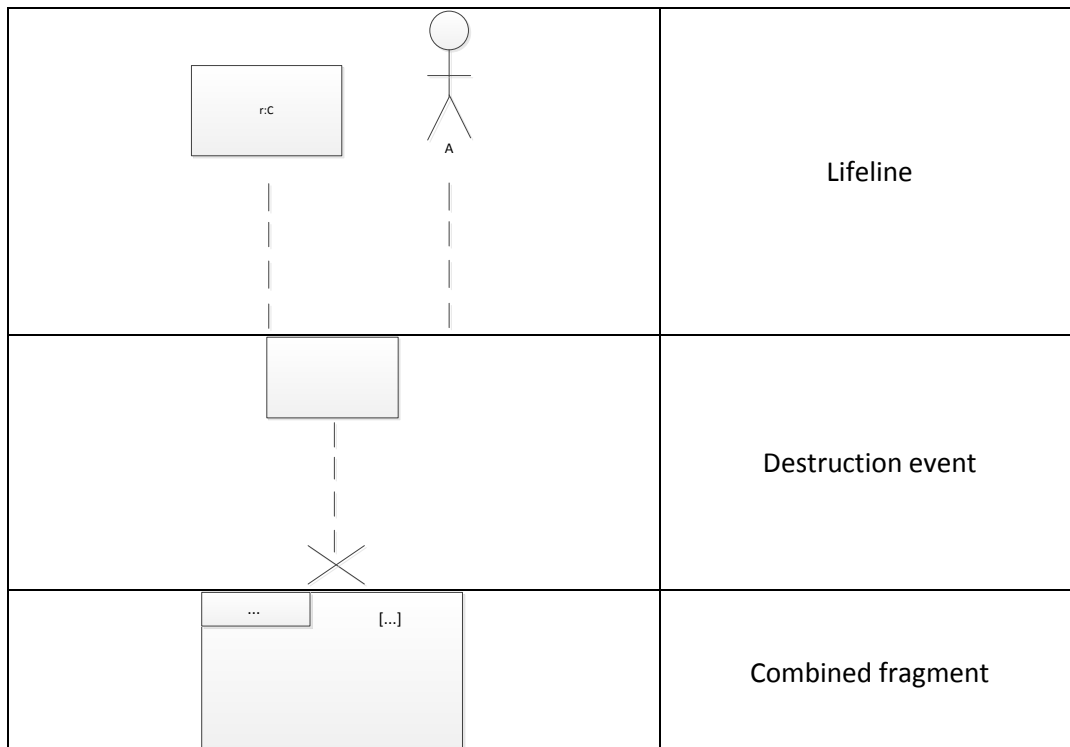
Kuvio 8 Aktivitettikaavio 2

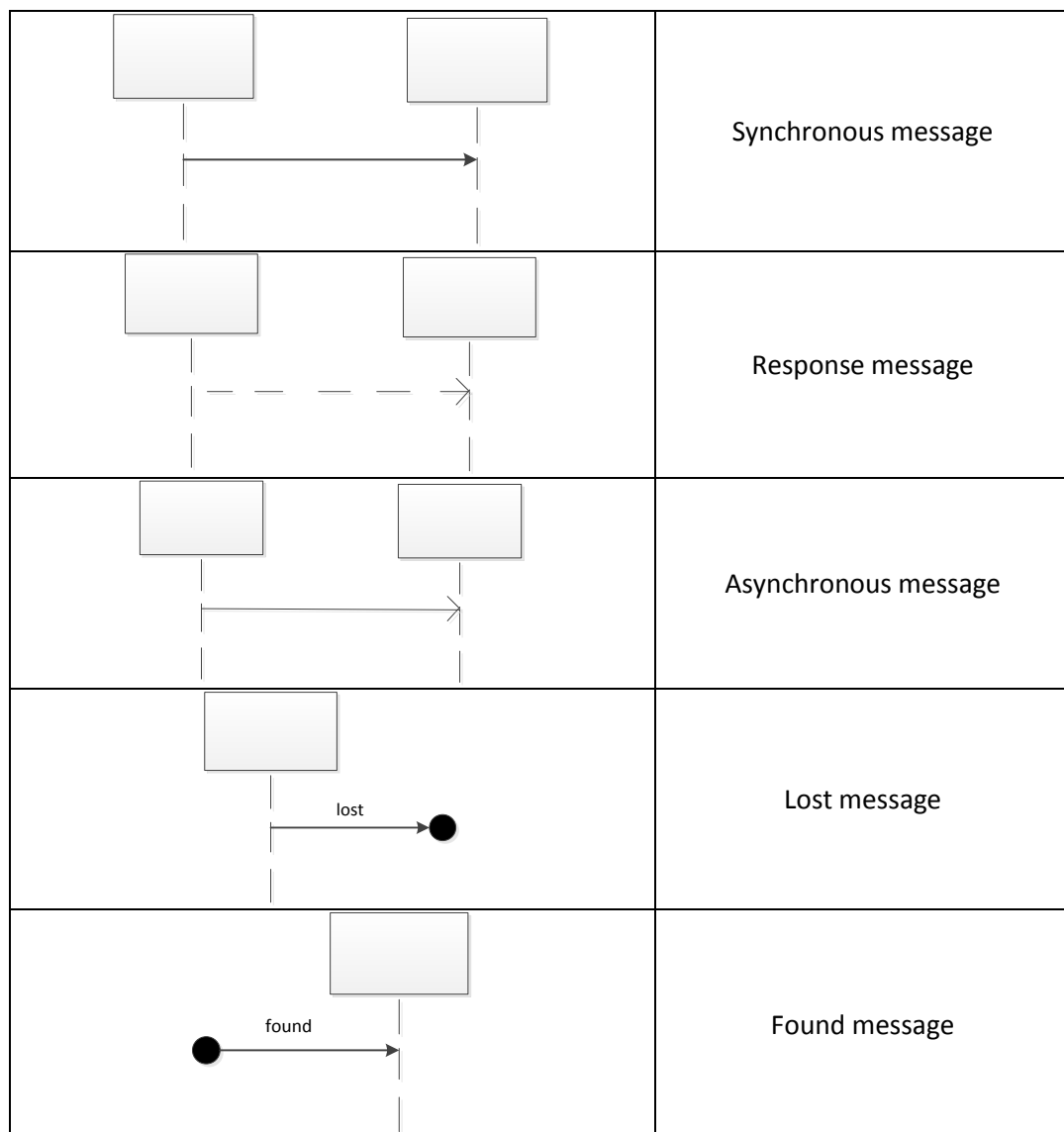
4.9 Sekvenssikaavio

Sekvenssikaavio(sequence diagram) on UML-kaaviotyyppi, jolla pystytään mallintamaan olio-ohjelmoinnissa olevien olioiden välisiä viestinvaihtoja (Haikala & Mikkonen 2011, 97). Tällä kaaviolla kuvataan tilakaaviossa tutuksi tulleiden tilakoneiden välistä kommunikointia. (Haikala & Märijärvi 2004, 152).

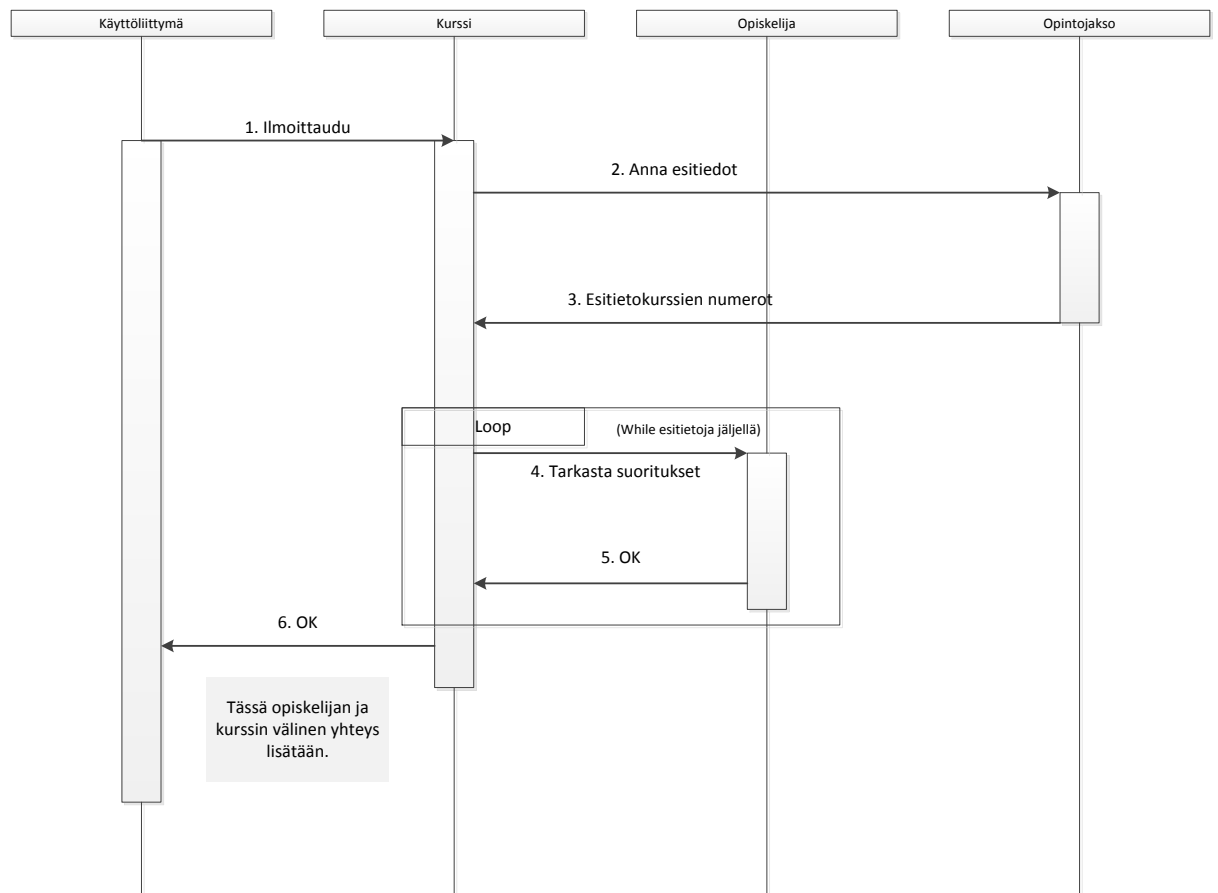
Sekvenssikaaviossa tapahtumaketju alkaa kaavion yläreunasta ja etenee vaiheittain kohti alareunaa. Viestit, jotka kaaviossa ovat alempana, tapahtuvat aina ylempien viestien jälkeen, mutta jos viestit ovat samalla kohdalla ne tapahtuvat samanaikaisesti. Kyseisessä kaaviossa moduulit asetetaan pystyakseleille ja näiden välinen viestinvaihto ilmaistaan nuolilla. Sekvenssikaaviolla pystytään myös kuvaamaan kuluva aikaa ja tapahtumisjärjestystä. (Kasurinen 2013, 40.)

Sekvenssikaavioita kutsutaan joskus myös skenaarioiksi. Skenaario on yleinen nimike tapahtumasarjan kuvaukselle ja sekvenssikaavioilla skenaarioiden kuvaus on yksi mahdollinen kuvaustapa. Täytyy muistaa, että aina mittavakaan kaaviojoukko ei riitä kuvaamaan kaikkia mahdollisia tapauksia, joten sekvenssikaavio yksinään ei aina ole riittävä spesifikaatio vaan enemmänkin kyseessä on havainnollinen kuvaus normaalitilanteessa tapahtuvista toiminnoista. Yksi tapa onkin mallintaa muutamia tyypillisiä tapahtumaketjuja ja muut tapaukset mallinnetaan sanallisesti. (Haikala & Märijärvi 2004, 152-153.) Kuviossa 9 on esitelty sekvenssikaaviossa käytettäviä symboleja.





Kuvio 9 Sekvenssikaavion symboleja (Seidl ym. 2015, 140)



Kuvio 10 Sekvenssikaavio (Haikala & Mikkonen 2011, 98)

Kuviossa 10 on mallinnettu sekvenssikaavio, jossa toiminta alkaa, kun käyttöliittymän kautta tulee ilmoittaudu funktion kutsu kurssi-oliolle. Ilmoittaudu funktion kutsun jälkeen, kontrollipalkki havainnollistaa aliohjelman sisällä olevan kontrollin. ”Ilmoittaudu” kutsuu opintojakson ”anna esitiedot” -metodia. Kurssi tarkistaa, että opiskelija, joka varausta yrittää suorittaa on suorittanut esitietokurssin. Kuvassa näkyvä loop on kontrollirakenteen operaattori, joka on sijoitettu kuvan sisällä olevan laatikon vasempaan ylänurkkaan ja kontrollirakenteeseen liittyvät ehdot esitetään hakasulkeissa. Jos esitiedot ovat kunnossa, linkki lisätään opiskelijasta kurssiin ja opiskelija on lisätty onnistuneesti kurssille. (Haikala & Mikkonen 2011, 98-100.)

5 LAADUN JA TIEDON OIKEELLISUUDEN MERKITYS JÄRJESTELMÄSSÄ

Tässä osassa käydään läpi laadun teoriaa, määrittelyä, varmistusta, mittausta sekä laatujärjestelmää. Tarkoituksena on avata lukijalle ymmärrys laadun merkityksestä ohjelmistotuotannossa.

5.1 Laatu

Laadunhallinta termissä esiintyvä sana laatu ei automaattisesti tarkoita hyvää laatua vaan sillä tarkoitetaan yleisesti kontrolloitua laatua (Haikala & Mikkonen 2011, 97). Tarpeesta johtuen laatu voi tarkoittaa esimerkiksi asiakkaan kokemaa laatua tai tuotteen teknistä laatua. Ohjelmistoprojekteissa ajatuksena on pyrkiä parhaaseen mahdolliseen laadulliseen lopputulokseen. Tarkoituksena on ensin kehitystyöllä toteuttaa paras mahdollinen ohjelma ja testaamalla varmistetaan järjestelmän täyttävän kaikki sille asetetut laatuvaatimukset. (Kasurinen 2013, 133.)

Laadun piiriin kuuluu useita erilaisia toimenpiteitä ja termejä. Laatuvaatimus, laadunvarmennus, laadunvalvonta, laatupäällikkö, laatusertifikaatit ja laatumalli ovat ohjelmistoprojekteissa yleisiä termejä, jotka viittaavat projektin tapahtumiin tai toteutettavalle tuotteelle määriteltyihin ominaisuuksiin. (Kasurinen 2013, 135.)

Alla kuvaus laadun piiriin kuuluvien termien määrittelyistä:

- Laatuvaatimukset asetetaan kehitettävälle järjestelmälle. Laatuvaatimusten tulee täytyä viimeistään hyväksymistestauksessa.
- Laadunvarmennuksessa arvioidaan kehitettävän järjestelmän toiminnot ja varmistetaan, että toiminnot vastaavat tehtyjä määrittelyjä.
- Laadunvalvonnassa varmistetaan, että tehtävä ohjelmisto valmistuu suunnitelman mukaisesti ja, että ohjelmisto tehdään oikein.
- Laatupäällikön tehtävänä on johtaa laadunvalvontaan liittyvää työtä.
- Laatusertifikaatteja seuraamalla varmistetaan palveluiden, toimintojen ja tuotteiden korkealaatu.

- Laatumalli on yrityksen sisäinen laadunvalvonnan ja laadunvarmennuksen toimintatapa. (Kasurinen 2013, 135.)

5.2 Laadun määrittely

Laatua voidaan määritellä kuudella perusmerkityksellä:

- Määrittelemätön laatu on käyttäjän tunne tuotteen korkeasta laadusta.
- Laatu on joukko ei mitattavissa olevista asioista tuotteessa.
- Tuotepohjainen laatu on laatua, joka perustuu määriteltyihin mittareihin.
- Käyttäjäpohjainen laatu on määritelmä, jossa tuote täyttää sille asetetut vaatimukset ja toimii vaatimusmäärittelyiden mukaisesti.
- Valmistuspohjainen laatu on laatu, joka pystytään todentamaan tarkastamalla, että tuote on luotu tarkasti ja korkealaatuisista osista ja järjestelmässä olevien virheiden määrä on pieni.
- Arvopohjainen laatu on tuotteen laatumääritelmä, jossa laatu perustuu tuotteen toiminnalliseen arvoon. (Kasurinen 2013, 136-137.)

ISO-standardeissa määritellään laatu siten, että laadukas tuote on sellainen, jolle asetetut toiminnalliset ja ei-toiminnalliset vaatimukset täyttyvät. Toiminnalliset vaatimukset ovat erilaiset toiminnot ohjelmistossa. Ei-toiminnalliset vaatimukset puolestaan ovat järjestelmän ominaisuuksia, kuten esimerkiksi luotettavuus. (Haikala & Märijärvi 2004, 193-194.)

5.3 Laatujärjestelmä

Laatujärjestelmäksi kutsutaan tuotetta, jonka luonnissa on käytetty yrityksen toimintatapaa ja jonka tavoitteena on saada tuotantoprosessi tuottamaan budjetin ja aikataulun mukaisesti etukäteen suunniteltuja laatutason tuotteita. Laatujärjestelmän toimivuus ja sen mukaan toimiminen tulee pystyä todistamaan tarpeen vaatiessa. Tämän vuoksi järjestelmän toiminnasta tulee jäädä todisteita, kuten esim. laadun mittaus tietoja tai virheraportteja. (Haikala & Märijärvi 2004, 48-50.)

Laatujärjestelmien arviointia pystytään tarkastelemaan erilaisten kehitettyjen arviointimallien avulla. Ohjelmistotuotannossa yleisin on ISO 9001 -standardi malli.

Yrityksen luomalle laatujärjestelmälle voidaan hakea ns. laatusertifikaattia, jolla yritys pystyy todistamaan järjestelmänsä olevan standardin mukainen, mutta laatusertifikaatin myöntäminen ei kuitenkaan takaa, että järjestelmä olisi täydellinen. (Haikala & Märijärvi 2004, 50.)

Laatujärjestelmään liittyy erilaisia standardeja, kuten ISO 9001. ISO 9001 standardi on toimialariippumaton järjestelmästandardi, jonka keskeisin tavoite on vähentää ulkoisten auditointien tarvetta. Ajatuksena on, että ulkopuolinen taho, joka on ISO:n valtuuttama, auditoi eli sertifioi sertifikaattia hakevan laatujärjestelmän. (Haikala & Mikkonen 2011, 146.)

Sertifioinnilla varmistutaan seuraavista seikoista:

- Sertifikaattia hakevalla on dokumentoitu laadunhallintajärjestelmä, joka täyttää ISO 9001 –standardin määrittämät vaatimukset.
- Sertifikaattia hakeva pystyy todentamaan, että toimii laadunhallintajärjestelmän mukaisesti.
- Sertifikaattia hakeva parantaa ja kehittää toimintatapojaan aktiivisesti. (Haikala & Mikkonen 2011, 146.)

ISO 9001 –standardi asettaa laadunhallintajärjestelmälle minimivaatimuksia, joiden sisältö voidaan yksinkertaisesti kiteyttää seuraavasti:

- Asetetaan selkeät tavoitteet toiminnalle.
- Dokumentoidaan toimintatavat, joilla päästään tavoitteisiin.
- Varaudutaan todistamaan, että yhtiössä toimitaan dokumentoitujen tapojen mukaan.
- Varaudutaan osoittamaan, että olemassa olevia toimintatapoja parannetaan ja kehitetään. (Haikala & Mikkonen 2011, 146.)

5.4 Laadunvarmistus

Oleellinen osa laatujärjestelmän toimintaa on laadunvarmistus. Laadunvarmistuksessa tarkastellaan tuotteen sekä toiminnan tekemisessä syntyviä vaihetuotteita. Toiminnan laatua voidaan tarkastella erilaisilla arviointitavoilla kuten auditoin-

neilla, jossa järjestelmä tai osa siitä käydään järjestelmällisesti läpi todentaen, että toiminta on laatujärjestelmän mukainen. Tuotteen laadunvarmistuksen tarkoituksena on estää tuotteisiin syntyviä virheitä ja löytää mahdollisimman aikaisessa vaiheessa tehtyjä virheitä. (Haikala & Märijärvi 2004, 51.)

Laadunvarmistusta pystytään tekemään mm. testaamalla. Laadunvarmistuksessa nousevat useasti esiin termit verifiointi ja validointi eli todentaminen ja kelpoistaminen. Verifiointissa tarkastetaan, että tuotetta rakennetaan oikein, kun taas validoinnissa tarkastetaan, että tuotetta rakennetaan oikeaan käyttötarkoitukseen. Tuotteen tarkastuksen tarkoituksena on vähentää vikoja mahdollisimman paljon jos sen syntyvaiheessa, jolloin seuraaviin vaiheisiin on helpompi siirtyä. (Haikala & Märijärvi 2004, 51.)

5.5 Laadun mittaaminen

Peruslähtökohtana yrityksissä on laatujärjestelmän kehittäminen paremmaksi ja tässä kehitystyössä mittaaminen on keskeisessä asemassa. Ilman laadun mittausta kehittämistoimenpiteiden tarpeita ja vaikutuksia on mahdotonta arvioida. Mittaukset kohdistuvat tuotteisiin ja tuotantoprosesseihin. Mittaaminen aiheuttaa lisäkustannuksia, koska siihen tarvittavan tiedon analysointi ja keräys vievät työtunteja. (Haikala & Märijärvi 2004, 204.)

Mittaamisen avulla pystytään löytämään ongelmakohdat ja kehitystarpeet prosessissa. Mittarit voidaan jakaa kahteen luokkaan, jotka ovat strategiset ja operatiiviset mittarit. Strategiset mittarit ovat liiketoimintanäkökulmasta tehtyjä pitkän aikavälin mittauksia. Lähtökohtana strategisissa mittareissa on ns. tasapainotettu mittaristo, joka sisältää asiakasnäkökulman, taloudellisen näkökulman, sisäisten prosessien näkökulman sekä kasvu- ja oppimisenäkökulman. Operatiivisista mittareista saadaan päivittäistä tietoa tukemaan päätöksentekoa. (Haikala & Mikkonen 2011, 143.)

Ohjelmistokehityksessä mittareilla on tärkeä rooli. Ilman mittauksia ei ole helposti tiedostettavissa, miten hyvin prosessi ja tuote toimivat, miten hyvin tavoitteisiin on päästy tai miten prosessia tai tuotetta tulisi kehittää. Mittaaminen ei ole help-

poa, mutta se on välttämätöntä. Osa mittareista voi olla todella tarkkoja, kun taas osa mittareista on vaikeita mitattavia. Mittaaminen kannattaa aloittaa mittaustavoitteista ja pyrkiä löytämään tavoitteiden tukemiseen sopivat mittarit. Huomioon tietenkin on otettava yrityksessä olevan laatu järjestelmän tila ja tarkastaa, millaisia mittareita on mahdollista toteuttaa kyseisessä kehitysvaiheessa. Sopivia mittareita ovat asiat, jotka ovat ongelmallisia ja asiat, joista kerätään ennestään mittausta mahdollistavaa dataa. Näiden avulla pystytään valitsemaan alustava mittarijoukko. (Haikala & Märijärvi 2004, 204.)

Mittaaminen kohdistuu yleensä prosessiin, projektiin tai tuotteeseen. Luotettavia yksittäisiä mittareita ei käytännössä ole ja mittareiden arvojen tulkintakin voi olla vaikeaa. Näitä arvoja tutkittaessa kannattaakin tarkastella useita arvoja rinnakkain. (Haikala & Mikkonen 2011, 143.)

Mittareiden käyttöön liittyy helposti myös ennakkoluuloja ja ihmiset voivatkin vastustaa mittareiden käyttöönottoa, koska ihmiset pelkäävät yksilön suorituksen mittauksissa esimerkiksi tulosten väärinkäyttöä. Syynä voi olla myös epäusko mittausta kohtaan, jolla voidaan tarkoittaa esim. epäilystä siitä, että mittauksia ei käytetä mihinkään. Tämän vuoksi mittaukseen liittyvät asiat tulisi avoimesti käsitellä ja mittaustulosten käytötavat tulisi olla tiedossa kaikille. (Haikala & Mikkonen 2011, 143.)

Mittareilta toivotaan ainakin seuraavia ominaisuuksia:

- Objektiivisuus ja luotettavuus, jolla tarkoitetaan että mittarin tulokset ovat mittaustilanteesta ja mittaajasta riippumattomia sekä mittausten pitää olla toistettavissa.
- Ymmärrettävyys, jolla tarkoitetaan, että mittareiden mahdolliset arvojen muutokset tulee olla selvillä.
- Asiakasnäkökulma, jolla tarkoitetaan, että osan mittareista tulee olla asiakkaidenkin ymmärrettävissä.
- Kustannukset, jolla tarkoitetaan, että mittaamisen tulee olla mahdollisimman automaattista.

- Taloudellinen merkitys osoitettavissa, jolla tarkoitetaan, että mittareiden arvot täytyy pystyä muuttamaan rahaksi.
- Mittaa prosessia ja ryhmää, ei yksilöä, jolla tarkoitetaan, että mittareilla ei kannata mitata esim. henkilökohtaisia palkkabonuksia.
- Mittarin väärinkäyttö on vältettävissä. (Haikala & Mikkonen 2011, 143.)

6 OHJELMISTOROBOTIIKKA TIETOTYÖSSÄ

Automatisoinnin kehittäminen oli tässä opinnäytetyössä keskeisessä osassa. Tässä kappaleessa perehdytään ohjelmistorobotiikkaan, joka on uudenlainen tapa tehdä automatisointia ohjelmistoissa. Tarkoituksena on saada lukijalle ymmärrys uudeltaisesta tavasta tehdä tietotyön automatisointia.

Ohjelmistorobotiikka eli RPA (Robotic Process Automation) on ohjelmisto tai ns. robotti, joka voidaan asettaa ottamaan tietokoneen ohjelmistot haltuunsa ja laittaa tulkitsemaan näitä ohjelmistoja tapahtuman käsittelyä varten (Institute for robotic process automation, 2016). Täytyy kuitenkin muistaa, että järjestelmissä olevan tiedon täytyy olla laadukasta, jotta ohjelmistorobotit pystyvät sitä käyttämään. (Kolehmainen, 2016).

Ohjelmistorobotiikalla on kaksi erillistä sovellustasoa. Ensimmäinen sovellustaso on nimeltään Robotic Process Automation (RPA), joka on ns. robotti, joka voidaan määrittää tekemään tiettyä sarjaa. Toinen sovellustaso on Smart Process Automation (SPA), joka on kehitetty monimutkaisempaan prosessiautomaatioon. RPA:n roboteista SPA:n robotit eroavat kyvyllään oppia. Oppimisen avulla SPA:n robotit pystyvät käsittelemään uusia vastaan tulevia tilanteita ja poikkeuksia, mihin niitä ei ole alun perin määritelty. (Haikonen 2016.)

RPA soveltuu automatisointiin seuraavissa prosessitilanteissa:

- Prosessi on stabiili ja kuvattu.
- Prosessin mittarit on määritelty ja käytössä.
- Prosessi on sääntöpohjainen.
- Prosessin rakenteellinen data on käytössä.
- Prosessissa on korkea suoritusmäärä. (Haikonen 2016.)

RPA:n käyttökohteita ovat erilaiset prosessitehtävät, joissa samaa kaavaa toistetaan yhä uudelleen. Tällaisia prosessitehtäviä ovat esimerkiksi datan siirtäminen ja tarkastaminen järjestelmien välillä tai datan muokkaaminen tehtyjen sääntöjen

pohjalta. Konkreettisella tasolla tarkoitetaan esimerkiksi raporttien muodostamista eri tietolähteistä tai tilausten käsittelemistä. (Haikonen 2016.)

SPA:lla voidaan toteuttaa automatisointia samankaltaisissa prosesseissa, kuten RPA:ssa, mutta SPA:n ominaisuutena on pystyä käsittelemään vapaamuotoista informaatiota, kuten esimerkiksi sähköposteja. SPA siis soveltuu prosesseihin, joissa tarkoituksena on käsitellä vapaamuotoista informaatiota ja sen informaation perusteella voidaan päätellä seuraava vaihe, kuten asiakaspalautteiden käsittely tai joissa robotti seuraa ihmisen tekemiä toimintoja ja oppia tämän avulla erilaisten tilanteiden käsittely. Robotit tällöin rakentavat itselleen logiikkaa ja algoritmia. (Haikonen 2016.)

Ohjelmistorobotiikkaa pidetään työelämän mullistajana, joka nähdään ratkaisuna perustietotekniikan sekä sovellusten harmonisointi- ja integrointipaineisiin. Eri-laisten business prosessien automatisoinnissa perinteinen it-projekti ei pärjää RPA-menetelmälle. Säästöt ovat huomattavat rahassa, ajassa sekä vaivassa. (Tamminen, 2016.)

Ohjelmistorobotiikan käyttöönoton tulee olla perusteltua, jotta robotti soveltuu todella prosesseihin, joihin sitä suunnitellaan ja että sen käyttöönotosta on todella hyötyä. Tällaiselle hankkeelle tulee tehdä business case –analyysi. Analyysissa otetaan huomioon laskennalliset hyödyt sekä kuvaukset laadullisista hyödyistä. (Haikonen, 2016.) Eera yhtiö on tekemillään ohjelmistoroboteilla pystynyt todentamaan, että roboteilla on pystytty laskemaan kustannuksia ja virtaviivaistamaan prosesseja (Eera, 2015).

Ohjelmistorobotiikan avulla on pystytty todentamaan toiminnan laadun parane- mista, koska inhimilliset virheet vähenevät. Tämän lisäksi sovelluksissa erilaiset toiminnallisuudet laajenevat ja elinkaari pitenee. Sovelluskehittäjille robotiikka tuo rajattomia mahdollisuuksia monipuolistaa ja kasvattaa nykyisiä it-ympäristöjä. (Tamminen, 2016.)

Ohjelmistorobotiikan hyödyt CGI määrittelee seuraavasti:

- Rutiinitehtävissä työvoiman tarve vähenee → Kustannussäästöt

- Läpimenoaika prosesseissa nopeutuu
- Työntekijät pystyvät keskittymään asiakkaille lisäarvoa tuottaviin palveluihin → Asiakaspalvelu paranee
- Asiakastyytyväisyys paranee → Palvelu on nopeaa ja tasalaatuista
- Työtyytyväisyys paranee → Työntekijät pystyvät keskittymään rutiinitöitä mielekkäämpiin tehtäviin
- Ohjelmistorobotti työstää prosesseja 24/7. (CGI, 2016.)

7 OPINNÄYTETYÖN TOIMINNALLINEN OSUUS

Opinnäytetyön toiminnallisessa osuudessa on tarkoituksena käydä läpi laajakaistatilauksen toimitus- ja automaatioprosessi, perumisen nykyhetki, miten perutut tapaukset tunnistettiin ja mitkä tapaukset valittiin automatisoinnin piiriin. Lopuksi kerron kaavioiden luonnista ja sanallisten käyttötapausten teosta.

7.1 Opinnäytetyön tavoite

Toimeksiantaja halusi vähentää kiinteän verkon dokumentointijärjestelmään sekä verkossa olevien liittymien hallintajärjestelmään syntyviä tietosisältövirheitä sekä helpottaa operatiivisille osastoille nousevien peruttujen kuluttajalaajakaistatilausten manuaalista käsittelyä.

Laajakaistatilauksen peruutuksen automatisoinnista haluttiin kehittää täysin automatisoitu toiminne. Tahtotilana oli saada tehtyä peruutuksen toimenpiteet siten, että järjestelmiin jäisi se tilanne, joka oli ennen muutoksia. Tämän lisäksi ajatuksena oli, että toimitusvastaavien ei tarvitse käsitellä peruttua laajakaistatilausta manuaalisesti, jolloin inhimilliset virheet vähenevät.

Seuraavaksi kerron nykytilanteen laajakaistatilauksen toimitusprosessista ja sen automaatioprosesseista ja siitä, miten tällä hetkellä laajakaistatilauksien perumiset käsitellään. Tämän jälkeen kerron laajakaistatilauksen perumisen automatisoinnin kehitystarpeesta.

7.2 Tilauksen toimitus -ja automaatioprosessi

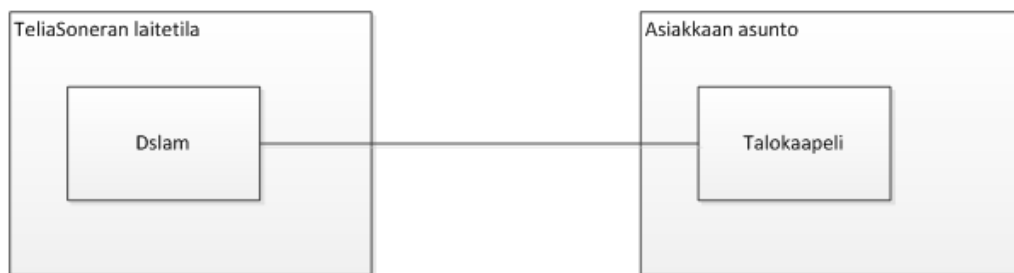
Uuden laajakaistatilauksen toimitusprosessi alkaa asiakkaan yhteydenotosta ja tilauksen tekemisestä. Asiakkaan tilatessa laajakaistaliittymän, tilaus tallennetaan tilausjärjestelmään, josta se siirtyy toimituksen hallintajärjestelmään. Tilauksen saapuessa toimituksen hallintajärjestelmään, tunnistetaan tilaus kuuluvaksi automaation piiriin, jota tässä opinnäytetyössä kutsutaan myöhemmin OrderFlowksi.

Orderflowssa määritellään, mitkä tilaukset kuuluvat automaation piiriin. Se sisältää seuraavat toiminnallisuudet:

- Automaattireitityksen verkon dokumentointijärjestelmään.
- Tunnistaminen mahdollisesta alihankkijasta ja luo alihankkijalle tarvittavat työrivit(Workflow).
- Tilauksella olevien rivien päivämäärien muuttaminen.
- Alihankkija varauksen varaus ja poisto tilaukselta ja JST-järjestelmästä.
- Sovittujen uusien rivien generoiminen.
- Sovittujen rivien kuittaus automaattisesti.
- Tilauksella olevien rivien siirtäminen oikeille työryhmille.
- Tilauksella olevien rivien ja tilauksien statuksien muuttaminen.

Toimituksen hallintajärjestelmä on yhteydessä kiinteän verkon dokumentointijärjestelmään sekä verkossa olevien liittymien hallintajärjestelmään. Kyseiset järjestelmät sisältävät tietoa mm. laajakaistayhteyksistä ja verkkolaitteista kyseisiin yhteyksiin liittyen.

Tilauksen siirtyessä OrderFlown piiriin, toimituksen hallintajärjestelmä ottaa yhteyttä kiinteän verkon dokumentointijärjestelmään ja automaatio lähtee etsimään vapaata reittiä tilauksessa ilmoitettuun osoitteeseen ja reitittää liittymän.



Kuvio 11 Näkymä kiinteän verkon dokumentointijärjestelmästä

Reitityksen tarkoituksena on saada muodostettua dokumentointi pisteestä A pisteeseen B eli Soneran verkkolaitteesta lähtevä kupariyhteys tulee saada kytkettyä asiakkaalle asti. Tämän jälkeen toimituksen hallintajärjestelmään siirtyy tieto siitä, mihin laitteeseen ja mihin porttiin yhteys on kytketty sekä erinäisten teletilojen ja laitteiden tiedot, joiden kautta yhteys saadaan asiakkaalle kytkettyä.

Automaattireitityksessä on erilaisia sääntöjä, joiden perusteella liittymä kuvataan verkon dokumentointijärjestelmään. Automaattireitityksen onnistuessa tulee siitä tieto toimituksen hallintajärjestelmään. Tämän jälkeen toimituksen hallintajärjestelmässä tarkistetaan automaattireitityksessä syntyneitä kytkentöjä, jonka perusteella luodaan tilaukseen mahdollisesti alihankkijalta tilattavat tuotteet. Tätä toiminnallisuutta automaatioprosessin sisällä kutsutaan Workflowksi.

Reitityksen jälkeen toimituksen hallintajärjestelmästä lähtee pyyntö konfiguroida liittymä verkkoon eli dokumentointijärjestelmästä tulleet tiedot kertovat, mihin verkkolaitteeseen ja verkkolaitteen porttiin liittymä tulee luoda. Toimituksen hallintajärjestelmästä lähtee alihankkijoiden järjestelmiin tieto asennuksesta, jos laajakaistaliittymän toimitus vaatii fyysisten kytkentöjen tekemistä.

Alihankkijan järjestelmässä tilauksen rivit siirtyvät työn alla tilaan. Alihankkijan järjestelmässä tapahtuvat tilamuutokset palautuvat toimituksen hallintajärjestelmään ja toimituksen hallintajärjestelmä lähettää tilatiedot myös tilausjärjestelmään. Asennusrivien yhteydessä asentajalle siirtyy toimituksen hallintajärjestelmästä tarvittavat kytkentätiedot, joiden perusteella asentaja käy fyysisesti tekevässä tarvittavat kytkennät. Kun asentaja on käynyt yhteyden kytkevässä, kuittaa hän tilauksen omalta osaltaan valmiiksi ja toimituksen hallintajärjestelmässä olevat asentaja rivit siirtyvät valmistilaan, jonka jälkeen koko tilaus kuittaantuu valmiiksi. Tilauksen kuittaannuttua valmiiksi toimituksen hallintajärjestelmässä lähtee tilauksen valmistumisesta kuittaus tilausjärjestelmään, jossa tilaus kuittaantuu valmiiksi ja siirtyy laskutukseen.

7.3 Laajakaistatilausten peruuminen nykyhetki

Tällä hetkellä toimitus järjestelmässä tilauksen peruutus toimii kahdella eri tavalla riippuen siitä, mihin vaiheeseen tilaus on toimituksen aikana edennyt. Tilaukseen kohdistuessa peruutus lähettävästä järjestelmästä, peruutuksen käsittelee joko automaatio -tai toimitusvastaava manuaalisesti.

Toimituksen hallintajärjestelmässä on automaatiotoiminne, jota tässä opinnäytetyössä oli tarkoituksena kehittää. Tällä hetkellä peruutuksen automaatio tarkistaa

tilauksella olevien rivien tilat, joiden perusteella automaatio käsittelee peruutuksen. Tilauksella olevilla riveillä voi olla yksi tila kerrallaan ja näitä tiloja on kuusi erilaista. Seuraavaksi avaan näiden rivien tilojen merkitystä ja kerron miten tilaus perutaan tämän hetkisen automatisoinnin toimesta.

UU = Työtä/riviä ei ole otettu käsittelyyn.

TO = Työ/rivi on siirretty työn alla tilaan.

LAH = Rivi odottaa vastausta toisesta järjestelmästä.

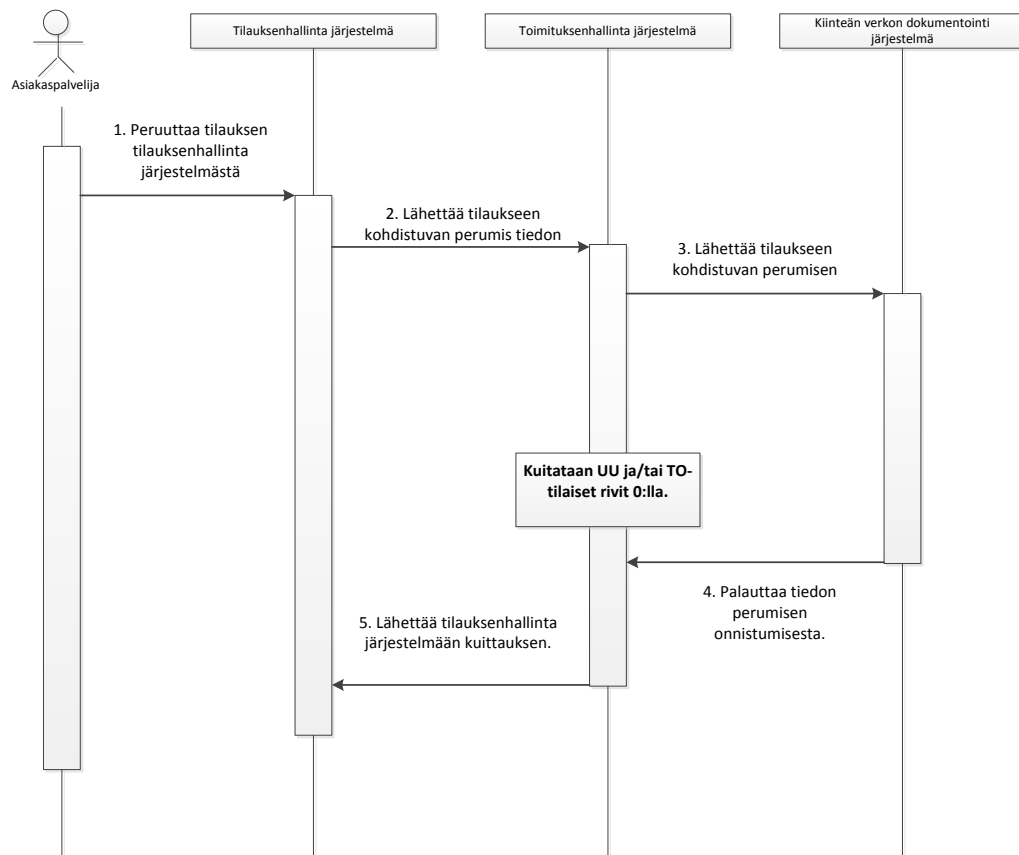
VA = Rivi on käsitelty ja valmistunut.

SII = Rivi on käsitelty ja siirtynyt valmiiksi.

EPA = Rivin käsittely on epäonnistunut

Käyttötapausten kuvausten yhteydessä muutettiin perumisen ehtoja rivien tilojen suhteen. Alkuperäinen ehto oli, että jos tilauksella olevien rivien tilat olivat UU-tilassa, niin tilauksella olevat rivit kuittaantuvat automaation toimesta 0:lla. Mikäli tilaukseen liittyi muita kuin UU-tilaisia rivejä, menee peruutuksesta tieto työn lisätietoihin sekä aliurakoitsijan tietojärjestelmään ja käyttäjä joutuu perumaan työn manuaalisesti.

Muutos koski tapauksia, joissa tilauksella olevien rivien tilat ovat UU- tai TO-tilassa. Tilauksella olevat rivit kuittaantuvat automaation toimesta 0:lla. Tällä muutoksella saatiin vähennettyä manuaalista käsittelyä tapauksille, joissa ei tarvinnut tehdä käänteisiä toimenpiteitä. Kuviossa 12 on mallinnettu sekvenssikaavio laajakaistatilauksen peruminen tilausrivien ollessa UU/TO-tilassa.



Kuvio 12 Laajakaistatilausten peruminen tilaus rivien ollessa UU/TO-tilassa

Jos tilauksella yhdellä tai useammalla rivillä on tilana LAH, VA, SII tai EPA, niin peruttu tilaus siirtyy manuaaliseen käsittelyyn, jolloin toimitusvastaavan tulee tehdä käänteiset toimenpiteet, jotta verkon dokumentointi saadaan palautettua siihen vaiheeseen, joka se oli ennen toimituksen aloittamista. Näitä käänteisiä toimenpiteitä ovat: verkon dokumentointijärjestelmään luodun reitin ja kytkentöjen poisto, kiinteän verkon hallintajärjestelmään konfiguroidun liittymän poisto sekä mahdollisesti tilatun vuokrajohdon irtisanominen.

7.4 Tapausten tunnistaminen perumisen automatisoinnin laajentamiseksi

Lähtökohtana oli valita automatisoidun peruutuksen piiriin tuotteet, jotka aiheuttivat eniten laajakaistan dokumentointi ja kiinteän verkon hallinta järjestelmiin tietosisältövirheitä. Aineistoa kerättyä lähdin tutkimaan, mitä tuotteista voitaisiin automatisoida.

Aineiston keräyksen osuudessa käytin hyväkseni, aiemmin mainitussa verkkotiedon oikeellisuuteen ja laatuun keskittyvässä projektissa tehtyjen korjausten aineistoa sekä projektin jälkeisessä operatiivisilla osastoilla tehtyjen korjauksien analysointeja. Korjausten yhteydessä oli kerätty aineistoon mm. analyysiä siitä, millaisesta virheestä oli kysymys ja juurisyyt, joista pystyin tarkastamaan, mistä virheitä syntyi, mihin järjestelmiin nämä virheet vaikuttivat sekä mistä tuotteesta oli kysymys. Peruttujen töiden tarkastelujen yhteydessä löytyi 20 erilaista versiota, joihin oli kohdistunut peruutus. Kuviossa 13 on esitetty tunnistetut tapaukset.

Tapaukset
Laajakaista ADSL(Oma alue)
Laajakaista ADSL(Toisen operaattorin alue)
Laajakaista valokuitu
Laajakaista ADSL:N Siirto/Purku(Omallalla alueella)
Laajakaista ADSL:N Siirto/Purku(Toisen operaattorin alueella)
Laajakaista ADSL:N Siirto/Purku(Omalta alueelta, toisen operaattorin alueelle)
Laajakaista ADSL:N Siirto/Purku(Toisen operaattorin alueelta, omalle alueelle)
Laajakaista ADSL:N Siirto/Purku(Omallalla alueella, vanhassa osoitteessa ADSL, uudessa osoitteessa valokuitu)
Laajakaista ADSL:N Siirto/Purku(Toisen operaattorin alueella, vanhassa osoitteessa ADSL, uudessa osoitteessa valokuitu)
Laajakaista ADSL:N Siirto/Purku(Omallalla alueella, vanhassa osoitteessa ADSL, uudessa osoitteessa valokuitu, työ aktivointi)
Laajakaista ADSL:N Siirto/Purku(Toisen operaattorin alueella, vanhassa osoitteessa ADSL, uudessa osoitteessa valokuitu, työ aktivointi)
Extran Inaktivointi(Vanhassa osoitteessa valokuitu, uudessa osoitteessa ADSL, joka omalla alueella)
Extran Inaktivointi(Vanhassa osoitteessa valokuitu, uudessa osoitteessa ADSL, joka toisen operaattorin alueella)
Laajakaista ADSL:N Vaihto/Purku (Omallalla alueella vanha tekniikka ADSL, uusi tekniikka valokuitu)
Laajakaista ADSL:N Vaihto/Purku (Toisen operaattorin alueella vanha tekniikka ADSL, uusi tekniikka valokuitu)
Laajakaista ADSL:N Vaihto/Purku (Omallalla alueella vanha tekniikka ADSL, uusi tekniikka valokuitu, vain aktivointi)
Laajakaista ADSL:N Vaihto/Purku (Toisen operaattorin alueella vanha tekniikka ADSL, uusi tekniikka valokuitu, vain aktivointi)
Laajakaistan Irtisanominen(Omallalla alueella)
Laajakaistan Irtisanominen(Toisen operaattorin alueella)
Nopeudenmuutos asentajakäynnillä
Nopeudenmuutos ilman asentajakäyntiä

Kuvio 13 Tunnistetut tapaukset

Tunnistettuja tapauksia lähdin tutkimaan avaamalla tilauksia ja tarkistelemalla tilauksella esiintyviä mahdollisia rivejä ja rivien tiloja eri toimituksen vaiheissa, jotka tulee huomioida perumisen automatisoinnissa. Kuvioissa 14, 15 ja 16 on esitettyä toimituksen hallintajärjestelmän mahdollisia näkymiä, kun tilaukseen on kohdistunut peruminen.

Kuviossa 14 on esitettynä näkymä toimituksen hallinnointijärjestelmästä, kun tilaus on peruttu. Kuviosta nähdään, että kaikkien rivien tilana on UU, joten tämän hetkisen määrittelyjen perusteella automaatio on kuitannut rivit 0:lla.

Rivin tila	Rivinro	Tuotenimi	Tila	Toimitettu
UU	1	LAAJAKAISTA ADSL	1	0
UU	2	SAATAVUUDEN VARMISTUS	1	0
UU	3	LAAJAKAISTA ADSL NOPEUS XM/XM	1	0
UU	4	LIITTYMÄN LUONTI VERKKOON	1	0
UU	5	UUSI YHTEYS: XXX-XXXXXX	1	0
UU	6	ALIHANKKIJATUOTE 1	1	0
UU	7	ALIHANKKIJATUOTE 2	1	0
UU	8	ALIHANKKIJATUOTE 3	1	0
UU	9	DATA-RIVI	1	0

Kuvio 14 Toimituksen hallinnointijärjestelmästä näkymä 1

Kuviossa 15 on esitettynä näkymä toimituksen hallinnointijärjestelmästä, kun tilaus on peruttu. Kuviosta nähdään, että Laajakaista ADSL ja Uusi yhteysriveillä on tilana TO ja muiden rivien tilana on UU. Tämän opinnäytetyön aikana käyttötapausten kuvausten yhteydessä muutettiin perumisen ehtoja rivien tilojen suhteen, joten uuden määrittelyn perusteella automaatio on kuitannut rivit 0:lla.

Rivin tila	Rivinro	Tuotenimi	Tila	Toimitettu
TO	1	LAAJAKAISTA ADSL	1	0
UU	2	SAATAVUUDEN VARMISTUS	1	0
UU	3	LAAJAKAISTA ADSL NOPEUS XM/XM	1	0
UU	4	LIITTYMÄN LUONTI VERKKOON	1	0
TO	5	UUSI YHTEYS: XXX-XXXXXX	1	0
UU	6	ALIHANKKIJATUOTE 1	1	0
UU	7	ALIHANKKIJATUOTE 2	1	0
UU	8	ALIHANKKIJATUOTE 3	1	0
UU	9	DATA-RIVI	1	0

Kuvio 15 Toimituksen hallinnointijärjestelmästä näkymä 2

Kuviossa 16 on esitettynä näkymä toimituksen hallinnointijärjestelmästä, kun tilaus on peruttu. Kuviosta nähdään, että useammalla rivillä esiintyy TO, VA ja LAH-tilat. Tämän hetkisen määrittelyjen perusteella automaatio ei pysty tilausta perumaan vaan toimitus vastaavan tulisi manuaalisesti käsitellä peruttu tilaus.

Rivin tila	Riviro	Tuotenimi	Tila	Toimitettu
TO	1	LAAJAKAISTA ADSL	1	1
VA	2	SAATAVUUDEN VARMISTUS	1	1
VA	3	LAAJAKAISTA ADSL NOPEUS XM/XM	1	1
VA	4	LIITTYMÄN LUONTI VERKKOON	1	1
TO	5	UUSI YHTEYS: XXX-XXXXXX	1	
LAH	6	ALIHANKKIJATUOTE 1	1	
LAH	7	ALIHANKKIJATUOTE 2	1	
LAH	8	ALIHANKKIJATUOTE 3	1	
VA	9	DATA-RIVI	1	1

Kuvio 16 Toimituksen hallinnointijärjestelmästä näkymä 3

Toiminnallisuuden tilaajan kanssa pidetyissä palaverissa kävimme läpi eri versiot tapauksista. Palaverissa päätettiin, mitkä tapauksista tulaisiin automatisoimaan. Kaikkiin tapauksiin täydellistä automatisointia ei haluttu, koska jo tunnistamisen vaiheessa havaittiin, että perumisen automatisoinnin logiikan luonti oli haastava ja riskialtis ja saatavat hyödyt verrattuna toteutuneisiin kustannuksiin eivät vastanneet toisiaan. Esimerkiksi siirtotapaukset jätettiin perumisen automatisoinnin ulkopuolelle.

Kaikkiin tuotteisiin täydellistä automatisointia ei haluttu, mutta kaikkiin tilauksiin, joihin kohdistuu peruutus tilausjärjestelmästä, generoituisi toimituksen hallinnointijärjestelmään uusi rivi nimeltään ”Peruutuksen toimenpiteet”. Kyseisen rivin lisätiedoissa on tarkoituksena olla käyttäjille muistutus perumisen toimenpiteistä. Kuviossa 17 automatisoinnin piiriin valitut tapaukset, joista tehtiin päätös 9.6.2015.

Tapaukset
Laajakaista ADSL (Oma alue)
Laajakaista ADSL (Toisen operaattorin alue)
Laajakaista valokuitu

Kuvio 17 Automatisoinnin piiriin valitut tapaukset

7.5 Perumisen automatisoinnin piiriin valitut tapaukset

Laajakaistatilauksen perumisen automatisoinnin piiriin tuli valituksi oman ja kilpailijan alueelle toimitettavat uudet ADSL -ja valokuitutekniikalla toimivat laajakaistaliittymät. ADSL-tekniikalla toimitettavissa laajakaistatuotteissa oman alueen

lisäksi toimitetaan kilpailijan alueelle laajakaistoja. Toimitettaessa laajakaista kilpailijan alueelle joutuu operaattori vuokraamaan vuokrajohtoa paikalliselta operaattorilta, jotta yhteys saadaan toimitettua tilaajalle asti. Vuokrajohtojen tilaukset johtuvat siitä, että Suomessa toimivat operaattorit on jaettu omiin verkkoalueisiin ja operaattoreilla ei ole suoraa pääsyä toisen operaattorin omistamiin laitteisiin ja kaapeleihin. Vuokrajohdolla toteutetut tapaukset tuottavat lisähaastetta perumisen automatisointiin, koska suurimpien operaattoreiden kesken on vuokrajohdon tilaus toteutettu toimitusjärjestelmien rajapintoja pitkin, joten perumisen automatisoinnissa tulee ottaa huomioon vuokrajohdon peruminen automaattisesti toisen operaattorin järjestelmään. Tämä vaatisi logiikka- ja rajapintamuutoksia järjestelmien välille.

7.6 Kaavioiden ja sanallisten käyttötapauksen luonti

Kaavioita lähdettiin kehittämään, kun automatisoinnin piiriin halutut tuotteet olivat selvillä. ADSL-tekniikalla toimivista laajakaistoista jouduin tekemään 3 erillistä kaaviota, koska kuten aiemmin mainitsin, oman alueen laajakaistatoimitusten lisäksi, operaattorit on jaettu omiin verkkoalueisiin ja operaattorista riippuen toimituksen hallintajärjestelmässä on erilainen automaatioprosessi vuokrajohdon tilauksessa.

Vaikka toimituksen alkuprosessi ADSL-tekniikalla toimitettavissa liittymissä oli sekä omalla että kilpailijan alueella hyvin samantyylinen, niin vuokrajohdon tilauksessa oli sekä automaatio että manuaalikäytäntöjä riippuen siitä, keneltä operaattorilta vuokrajohtoa tilattiin. Tämän vuoksi helpoin tapa oli luoda 3 erillistä kaaviota.

Jokaisesta tapauksesta kehitin oman kaavion, jolloin toimeksiantajan oli helppo ymmärtää, miten järjestelmä tulisi toimimaan missäkin vaiheessa. Toimittajan puolestaan on helpompi tehdä työmääräarviot sekä mahdollinen myöhemmän vaiheen koodaus ja testaus.

Sanalliset käyttötapaukset muodostuvat seuraavista kokonaisuuksista:

- Otsikko, kuvaava nimi tapaukselle.
- Yleiskuvaus, lyhyt kuvaus mitä tapahtumaa käyttötapaus kuvaa.
- Muutoshistoria, jossa ylläpidetään dokumenttiin tehdyt muutokset ja niiden päivämäärät.
- Esiehdot, tapahtumat/toiminteet, jotka oltava suoritettu ennen kuin ko. käyttötapauksista voidaan toteuttaa.
- Tekstikuvaus, tekstikuvaukseen kuvataan todennäköisin suorituspolku. Mitä halutaan saada aikaiseksi?
- Poikkeukset ja poikkeuksista selviytyminen, kirjataan esim. virhetilanteet ja kerrotaan miten niistä jatketaan.

Sanalliset käyttötapaukset pyrittiin kuvaamaan ymmärrettäviksi, jotta käyttäjät pystyvät ymmärtämään tapauksen. Vältettiin myös ottamasta kantaa toteutukseen ja pyrittiin tekemään kuvaukset siten, että ne toimivat testauksen apuna.

Kaavioiden ja sanallisten käyttötapauksien luonnissa helpotti suuresti, kun tunsin käytännön tasolla toimitusten hallinnointijärjestelmän, kiinteän verkon dokumentointijärjestelmän sekä verkossa olevien liittymien hallintajärjestelmän.

8 JOHTOPÄÄTÖKSET JA POHDINTA

Opinnäytetyön tavoitteena oli mallintaa automaation kehittämistä varten kaaviot ja luoda sanalliset käyttötapaukset työmääräarviota sekä myöhempää koodausta ja testausta varten. Kaavioiden ja sanallisten käyttötapausten luonti toteutettiin toimeksiantajan antamassa aikataulussa.

Haasteena oli saada huomioitua kaikki mahdolliset sanalliset käyttötapaukset. Kuten kappaleessa 7.4 esitinkin, niin toimituksen hallintajärjestelmässä tilaus riveillä on erilaisia tiloja ja niiden kaikki mahdolliset tilat tulee ottaa huomioon. Sanallisten käyttötapausten haasteesta kertookin mielestäni se, että sanallisia käyttötapauksia kirjoitin noin 200 sivu ja näiden lisäksi Microsoft Visiolla mallinsin 4 erilaista kaaviota.

Omasta mielestäni kaavioiden ja sanallisten käyttötapausten luonti onnistui hyvin ja aikataulullisesti. Kaavioiden ja sanallisten käyttötapausten tekemistä helpotti suuresti se, että osasin jo entuudestaan käyttää tilauksen hallinta-, toimituksen hallinta-, kiinteänverkon dokumentointi- sekä verkossa olevien liittymien hallintajärjestelmiä. Kokonaisuutena empiirinen osuus vastaa hyvin opinnäytetyön tavoitetta. Automatisointia olisi voinut kehittää ottamalla eri tuotteita automaation piiriin, mutta todennäköisesti kustannukset olisivat voineet estää toteutuksen.

Jatkokehitys ehdotuksena automatisointia voisi yrittää laajentaa muihin tuotteisiin tai tehdä tutkimuksen tässä opinnäytetyössä esitettävästä automatisoinnin vaikutuksista.

LÄHTEET

CGI, 2016. Automatisoi rutiinityöt robotin hoidettavaksi. Viitattu: 30.10.2016.
<https://www.cgi.fi/bi/ohjelmistorobotiikka>

Eera, 2015. Ohjelmistorobotiikka tuottavuusloikan lähteenä. Viitattu: 30.10.2016.
<http://www.eera.fi/ohjelmistorobotiikka-tuottavuusloikan-lahteenä/>

Enden J. 2000. UML:n perustekniikat ja Rational Rose Enterprise Edition – työkalu. Helsingin yliopisto.
 Viitattu [24.9.2016]. <https://www.cs.helsinki.fi/u/laine/otv/enden.pdf>

Haikala I & Mikkonen T. 2011. Ohjelmistotuotannon käytännöt. Helsinki: Talentum Media Oy.

Haikala I & Märijärvi J. 2004. Ohjelmistotuotanto. Helsinki: Talentum Media Oy.

Haikonen, M. 2016. Osa2: Ohjelmistorobotiikka – mitä se on? Viitattu: 29.10.2016.
<https://www.linkedin.com/pulse/osa-2-ohjelmistorobotiikkamist%C3%A4-siin%C3%A4-kyse-mika-haikonen?published=t>.

Haikonen, M. 2016. Osa 3: Ohjelmistorobotiikan hyötynäkökulmia. Viitattu: 29.10.2016.
<https://www.linkedin.com/pulse/osa-2-ohjelmistorobotiikkamist%C3%A4-siin%C3%A4-kyse-mika-haikonen?published=t>.

Harsu, M. 2003. Ohjelmien ylläpito ja uudistaminen. Helsinki: Talentum Media Oy.

Institute for robotic process automation, 2016. What is Robotic Process Automation? Viitattu: 29.10.2016. <http://www.irpanetwork.com/what-is-robotic-process-automation/>

Kasurinen, J. 2013. Ohjelmistotestauksen käsikirja. Jyväskylä: Docendo.

Kolehmainen A. 2016. Ohjelmistorobotit mullistavat työelämän – "tulee vastaava taito kuin Excelistä". Viitattu: 30.10.2016.
http://www.tivi.fi/Kaikki_uutiset/ohjelmistorobotit-mullistavat-tyoelaman-tulee-vastaava-taito-kuin-excelista-6537565

Sahi, A. 2010. Viitattu: 31.10.2016.
<http://www2.amk.fi/digma.fi/www.amk.fi/opintojaksot/0303012/1106227851022/1106577077518/1107020112812/1149490103657/toimintokaavioidensymboleidenkaytto.html>

Sanastokeskus TSK, 2012. Viitattu: 18.10.2016.
<http://www.tsk.fi/tiedostot/pdf/Laajakaistanasanasto>

Seidl M, Scholz M, Huemer C & Kappel G. 2015. UML @ Classroom: An Introduction to Object-Oriented Modeling. Switzerland: Springer Cham.

Sininen meteoriitti, 2013. Viitattu: 20.2.2016.
<http://www.meteoriitti.com/2013/06/06/ketteryys-haltuun-yleisimmat-ketterat-kaytannot/>

Tamminen O. 2016. Työelämää mullistava ohjelmistorobotti uurastaa väsymättä.
 Viitattu: 30.10.2016. [http://net.fujitsu.fi/fi-FI/12016/Tyoelamaa_mullistava_ohjelmistorobotti_u\(9656\)](http://net.fujitsu.fi/fi-FI/12016/Tyoelamaa_mullistava_ohjelmistorobotti_u(9656))

Telia Company, 2016. Viitattu: 6.11.2016.
<http://www.teliacompany.com/en/newsroom/news/news/news-articles/2016/5q-about-5g>

TeliaSonera Finland Oyj, 2016. Viitattu: 21.10.2015.
<http://www.sonera.fi/teliasonera>

TeliaSonera kotisivut, 2013. Viitattu: 18.7.2015. <http://www.sonera.fi/teliasonera>

Tietoliikenteen ja tietotekniikan keskusliitto, FiCom ry. 2016. Viitattu: 21.10.2016. http://www.ficom.fi/tietoa/tietoa_4_1.html?Id=1254998524.html

Wikispaces, 2016. Viitattu: 14.11.2016. <http://banking-system.wikispaces.com/file/view/stat2.jpg/245514829/stat2.jpg>